Universidad Carlos III de Madrid

Escuela Politécnica Superior

Grado en Ingeniería de Sistemas de Comunicaciones

Plan 2008

Trabajo de fin de grado

StarCrush: diseño y desarrollo de un juego tipo "conecta-tres" de lógica y preguntas educativas en Python y Pygame.

Autor: **Enrique Martín Hernández**

Tutora: **Iria Estévez Ayres**

Septiembre de 2014

Agradecimientos

Empiezo esta ronda de agradecimientos acordándome, como no podía ser de otra forma, de mis familiares, de todos ellos, que siempre creyeron en mí, hasta cuando yo mismo dudaba.

Continúo nombrando a mis amigos. A los de toda la vida, que siempre han estado conmigo y confío en que siempre estarán. Y obviamente quiero agradecer también, a los amigos que he conocido en la carrera, a los que han perdurado a lo largo de la misma y los que espero que no desaparezcan nunca de mi lado.

Todos ellos, familiares y amigos me han llevado en volandas hasta la recta final, dándome aliento y ánimo cuando más falta me hacía y consejo cuando más necesario era.

Por último, quiero dar mi agradecimiento a todos los profesores con los que me he cruzado en la carrera. En especial a Iria, mi tutora de este trabajo de fin de grado. Muchas gracias por ayudarme en todo, por tus consejos, paciencia y dedicación.

Es de bien nacidos ser agradecidos.

Resumen

En la presente memoria de trabajo de fin de grado se explica de forma detallada el proceso de análisis, diseño e implementación de una aplicación con fines docentes. En concreto, la aplicación se trata de un juego del tipo conecta-tres y pretende ser usada como herramienta de apoyo para el aprendizaje de una asignatura.

Para tal fin se pretende conseguir que los alumnos se vean realmente incentivados a jugar y de forma indirecta estarán aprendiendo el contenido teórico de dicha asignatura, ya que el juego incluye un sistema de recuperación de vidas basado en preguntas con contenido de la asignatura que debe contestar el jugador.

La aplicación se ha realizado como una aplicación de escritorio, mediante el lenguaje de programación Python y el motor de videojuegos Pygame. Además hace uso de una base de datos, implementada mediante SQLite, para guardar los datos relativos al jugador, desde su nombre y contraseña hasta su progreso en el juego.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

En este primer capítulo introductorio se explica la motivación que ha llevado al desarrollo del presente trabajo, así como los objetivos que con él se quieren conseguir. Además se explica el marco regulador que afecta a la aplicación desarrollada. Por último, también se detalla la estructura de la presente memoria.

1.1. Motivación

En la primera parte de la introducción se presentan los principales aspectos que han motivado la implementación de este trabajo de fin de grado.

El gran número de alumnos que abandona la evaluación continua correspondiente a las asignaturas de programación es un hecho muy significativo. Esta realidad puede interpretarse como una dificultad añadida a la que se ven sometidos los alumnos al enfrentarse por primera vez a dichas asignaturas.

La aplicación que se ha desarrollado en el presente trabajo de fin de grado pretende hacer frente a esta realidad y reducir dicha dificultad. Para ello, se ha pensado en un videojuego como herramienta educativa. Se aprovechará el carácter competitivo de dicho videojuego para incentivar su uso por parte del alumno, de forma que aprenderá el contenido teórico de la asignatura a medida que juega.

1.2. Motivación personal

En este apartado, se ha querido reservar un hueco para hablar de la motivación personal que me ha empujado a realizar el presente trabajo de fin de grado.

En primer lugar, la decisión de hacer un trabajo de fin de grado ambientado en el ámbito de la programación surge por un gusto personal por este sector. Programar puede ser la tarea más desesperante que he realizado a lo largo de la carrera, pero finalmente, cuando todos los problemas se han solucionado y todo ha salido como se esperaba, me ha proporcionado algunas de las mayores alegrías, a parte de una sensación de superación fantástica de experimentar. Una sensación que en ocasiones hace que te sientas como un mago, haces algo que todos pueden ver, pero pocos saben como lo has hecho.

Además la motivación por programar en el trabajo de fin de grado, también se ve incentivada por la sensación de no haberlo hecho lo suficiente a lo largo de la carrera, por no ser demasiadas las asignaturas orientadas a tal fin, por lo que he considerado que era una buena oportunidad para reforzar los conocimientos adquiridos y ampliarlos considerablemente.

1.3. Objetivos

A continuación se exponen los objetivos que se quieren conseguir con el presente trabajo de fin de grado y que como es natural surgen de la propia motivación, explicada anteriormente, que da lugar a la realización del mismo.

El principal objetivo de este trabajo de fin de grado ha sido realizar una aplicación, basada en un videojuego en 2D, con diferentes niveles de corta duración y un mecanismo de vidas.

La aplicación debe permitir al usuario recuperar vidas para poder seguir jugando, lo cual debe conseguirse mediante la respuesta de preguntas de tipo test con contenido de la asignatura.

Además la aplicación debe ser capaz de guardar la evolución del jugador, para que pueda jugar al ritmo que quiera y no pierda lo conseguido. Por ello, la aplicación hará uso de un sistema de gestión de bases de datos, para poder guardar un registro de cada jugador, al cual sólo podrá acceder él mismo por medio de un nombre de usuario y una contraseña. Esto es muy importante, ya que a nadie nos gustaría que los demás jugasen nuestras partidas.

Además, la aplicación como se mencionó anteriormente debe contener niveles de distinta dificultad y no debe permitir que el jugador juegue al nivel que quiera. Por ello, haciendo uso de nuevo de la base de datos, permitirá al alumno únicamente jugar a los niveles que correpondan en cada caso.

En resumidas cuentas, las ideas que generan la motivación de realizar esta aplicación acaban por convertirse en los objetivos de la misma, que básicamente es la de un videojuego que pueda utilizarse como herramienta educativa aprovenchándonos de algunas carácterísticas propias a muchos videojuegos como son la capacidad de entretener y de competición.

Finalmente no puedo cerrar esta sección sin nombrar un objetivo que aunque no pertenece a la propia aplicación es fundamental para mí, pues es de carácter más personal. Es decir, este objetivo se aleja de los mencionados anteriormente y es independiente del tipo de aplicación a realizar, pues se trata de un objetivo simple pero fundamental y es el objetivo de aprender. Éste fue sin ninguna duda mi primer objetivo, el que hizo que me decantara por realizar el trabajo de fin de grado en el ámbito de la programación. Se trata de un campo que me apasiona y que consideraba importante reforzar, por lo que podría decirse que este objetivo fue el punto de partida en este camino y el desencadenante de objetivos posteriores.

En definitiva una enumeración de los principales objetivos es la siguiente:

- Estudio de las herramientas necesarias para el desarrollo de la aplicación.

- Elección de un lenguaje de programación, adecuado para el desarrollo de la aplicación, que no haya sido estudiado durante la carrera.
- Diseño de un prototipo de juego tipo conecta-tres.
- Desarrollo de un sistema de recuperación de vidas que convierta al juego en una herramienta de apoyo docente.
- Desarrollo de una base de datos que permita la lectura, escritura y modificación de los datos usados por la aplicación.

1.4. Marco regulador

En este apartado se habla acerca del marco regulador, en cuanto al manejo de datos se refiere, que afecta a la presente aplicación. Para ello es necesario mencionar la LOPD (Ley Orgánica de Protección de Datos). Se trata de una Ley Orgánica española que tiene por objeto garantizar y proteger, en lo que concierne al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y especialmente su honor, intimidad y privacidad personal y familiar. Su objetivo principal es regular el tratamiento de los datos y ficheros, de carácter personal, independientemente del soporte en el cual sean tratados, los derechos de los ciudadanos sobre ellos y las obligaciones de aquellos que los crean o tratan.

A partir de la aparición de esta ley se creó la Agencia Española de Protección de Datos, organismo que vela por el cumplimiento de esta normativa a nivel nacional.

Cuando se habla de ficheros se hace referencia al conjunto organizado de datos de carácter personal, cualquiera que sea la forma de su creación, organización y acceso. Aquél que cree este tipo de ficheros deberá inscribirlos en el Registro General de Protección de Datos (RGPD), informar y obtener consentimiento acerca del tratamiento o guardar secreto entre otras cosas.

Por otro lado, es interesante destacar que la ley establece tres niveles de medidas de seguridad: básico, medio y alto. Esta clasificación se realiza en función de la naturaleza de la información tratada en el fichero, en relación con la menor o mayor necesidad de garantizar la confidencialidad y la integridad de la información **[1]**.

A continuación se detalla el tipo de ficheros y el tratamiento de los mismos en función del nivel de seguridad al que pertenezca:

- **NIVEL ALTO:** se trata de ficheros o tratamiento con datos de ideología, afiliación sindical, religión, origen racial, salud o vida sexual.
- **NIVEL MEDIO:** se trata de ficheros o tratamiento con datos relativos a la comisión de infracciones administrativas o penales que se rijan por el artículo 29 de la LOPD (prestación de servicios de solvencia patrimonial y crédito), datos de Administraciones tributarias, de entidades financieras o de Entidades Gestoras y Servicios Comunes de Seguridad Social entre otros datos.
- **NIVEL BÁSICO:** se trata de cualquier otro fichero que contenga datos de carácter personal, tales como nombres y apellidos, códigos postales, domicilio, ciudad o teléfono.

Las medidas de seguridad de nivel básico son exigibles en todos los casos [2].

Vista la anterior clasificación, es evidente que en esta aplicación únicamente se hace uso de datos que corresponden al nivel más básico, ya que incluso el nombre de usuario, puede ser cualquiera y no el verdadero nombre del jugador.

Tal y como se ha mencionado anteriormente, el fichero debe ser inscrito en el RGPD. Para ello se realizará el registro asignando un nombre al fichero y un número identificativo. Este identificador es el número de alta cuando se inscribe un cierto fichero y de esta forma estará siempre asociado a la información que se desea proteger.

Además, una vez realizado el registro, se cumplimenta un documento que detalla exactamente el tipo de información que se debe proteger y con qué nivel de protección.

En la **Figura 1.1** se muestra el documento mencionado más arriba.

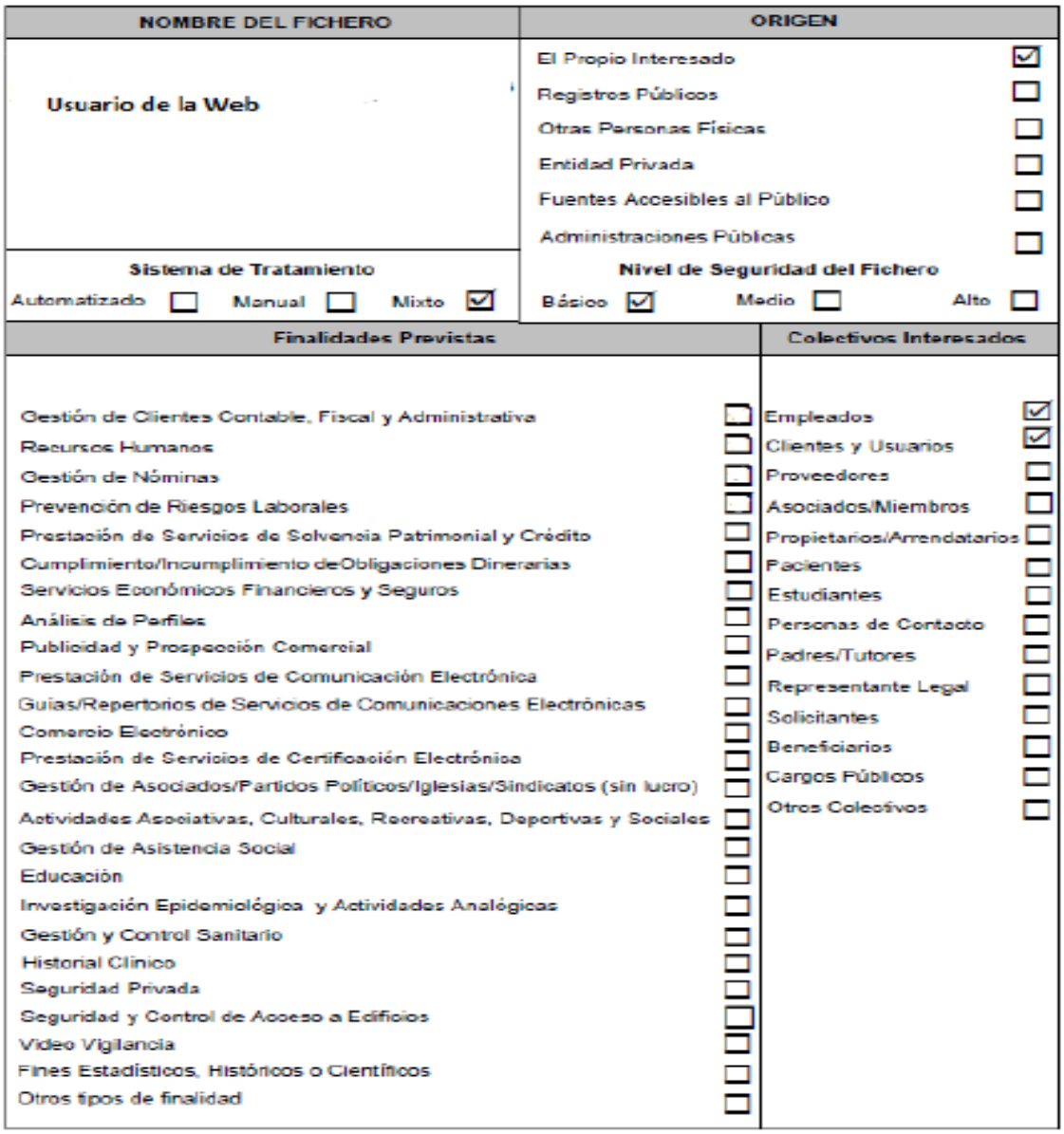

Figura 1.1: Documento de seguridad para ficheros de datos.

1.5. Estructura de la memoria

El presente documento recoge toda la información relacionada con la realización del trabajo de fin de grado y está dividido en diferentes capítulos. Cada uno de estos capítulos abarca una determinada información, relacionada entre sí y que describe un determinado aspecto del presente trabajo de fin de grado.

A continuación se describe brevemente la temática de la información recogida en cada capítulo, lo que se espera que sea de ayuda para una mejor comprensión futura de cada uno de ellos.

1. **Introducción:** en este capítulo se describe el contexto en el que se encuentra el desarrollo del trabajo de fin de grado, presentando las principales ideas del mismo de manera muy básica así como los principales objetivos que se desean obtener con el desarrollo de la aplicación.

2. **Estado del arte y elección de tecnologías:** en este capítulo se hace un repaso general a la historia de los videojuegos, desde sus inicios hasta la actualidad, centrándose especialmente en su relación con la educación. Además también en este capítulo se expondrán diferentes alternativas para llevar a cabo la aplicación, realizando un estudio de las tecnologías y lenguajes de programación existentes que podrían servir para la realización del presente trabajo y de esta forma poder elegir la combinación óptima.

3. **Diseño e implementación:** en este capítulo se describirá como se ha llevado a cabo el proceso de desarrollo de la aplicación, profundizando en aspectos más tecnicos. Se detallarán los requisitos que debe cubrir la aplicación para cumplir los objetivos. También se explicarán los diferentes módulos de los que consta la aplicación y que permitirán el entendimiento del flujo seguido por ésta.

4. **Validación de los requisitos:** en este capítulo se demostrará que la aplicación cumple los requisitos planteados en el capítulo anterior y que son fundamentales para cumplir los objetivos buscados en un principio. Para conseguirlo se hará uso de imágenes del propio videojuego y de una detallada explicación de cada caso.

5. **Conclusión final y trabajo futuro:** en este capítulo se presentan las conclusiones referentes a todo el proceso de desarrollo de la aplicación que conforma el presente trabajo de fin de grado. Además se presentan posibles desarrollos futuros, que supondrían o una mejora en la aplicación o una ampliación de la misma.

6. **Planificación y presupuesto:** en la parte final de la memoria se incluye un apéndice que consta de dos partes:

    - Planificación: en esta primera parte del apéndice se desglosa en tareas todo el trabajo realizado en el presente trabajo de fin de grado y se hace una estimación de la realización temporal de cada una de ellas.
    - Presupuesto: por otro lado, en la otra parte del apéndice se hace el presupuesto que supone llevar a cabo el trabajo de fin de grado, considerando tanto las horas dedicadas a cada tarea como el coste del material empleado en la realización del mismo.

Capítulo 2

Estado del arte y elección de tecnologías

En el presente capítulo se va a hablar de la historia de los videojuegos, concretando en su aplicación a la educación, así como su evolución y el porqué de su efectividad en la docencia.

También se van a analizar diferentes lenguajes de programación así como diferentes librerías, todas ellas aptas para el desarrollo de videojuegos. Esta diversidad de opciones es una de las primeras decisiones a las que se enfrenta el programador, por lo que también se explicará el motivo de la elección.

En definitiva, este capítulo explica la temática del presente proyecto y permite una mejor comprensión de los siguientes capítulos.

2.1. Los videojuegos y la educación

En primer lugar se va a hablar de la evolución de los videojuegos, su situación actual y su relación con el ámbito de la educación.

2.1.1. El nacimiento de los videojuegos

Hay que remontarse a los años 40 para poder hablar de los inicios de los actuales videojuegos. Es en esta época cuando los técnicos americanos desarrollaron el primer simulador de vuelo, con el fin del entrenamiento de los pilotos. El desarrollo continuado de los videojuegos tuvo lugar con la aparición de la tercera generación de ordenadores en el año 1962.

El nacimiento del microprocesador en 1969 supone un gran avance ya que consigue aumentar la potencia de información, constituyendo así el corazón de los ordenadores, videojuegos y calculadoras de hoy día.

Es en el año 1972 cuando se fija la aparación del primer videojuego, llamado PONG, que consistía en una rudimentaria partida de tenis o ping-pong. Y es en el 1977 cuando la firma Atari lanza al mercado el primer sistema de videojuegos en cartucho, la cual tuvo un gran éxito. Este hecho hizo que por primera vez se plantearan la influencia que

los videojuegos podrían tener en la conducta y en la educación de los niños.

A partir de aquí las constantes mejoras técnicas, como el aumento de la potencia de los microprocesadores o la memoria, permitió también la rápida evolución de los videojuegos, mejorando enormemente su calidad y llevando a cabo su extensión masiva en la década de los años 90. Dada la influencia que habían alcanzado se produjo una segunda investigación sobre los posibles efectos negativos de los videojuegos en la población, desde diferentes campos, como la medicina, la sociología, la psicología y la educación **[3]**.

En la **Figura 2.1** se muestra una visualización del que se considera el primer videojuego.

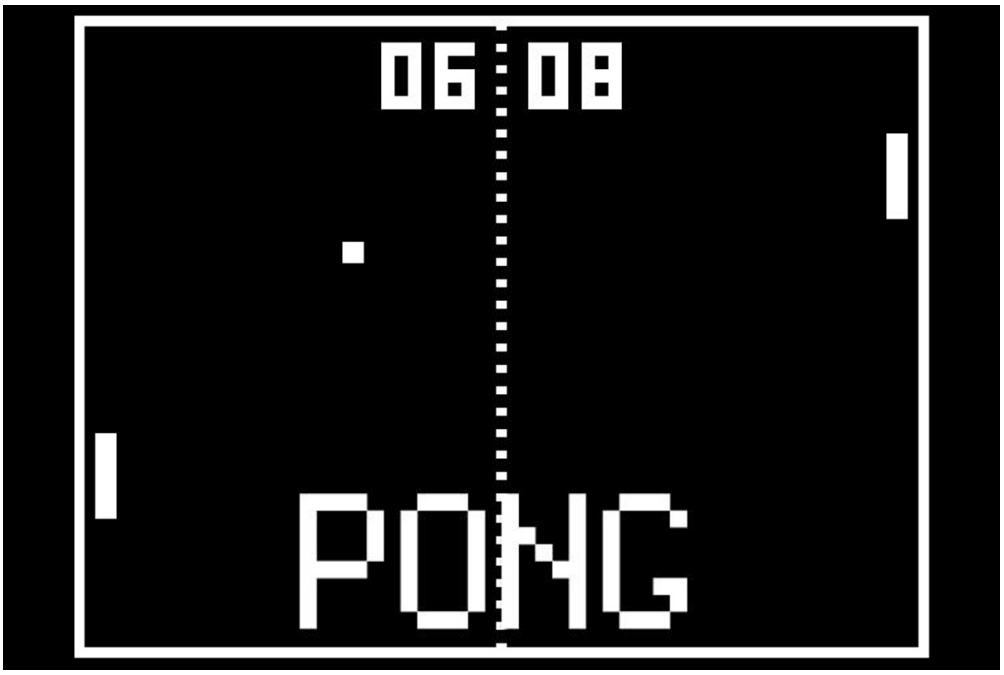

Figura 2.1: Videojuego PONG.
[Imagen tomada de la página www.tupiensas.com]

2.1.2. El éxito de los videojuegos

Algo que se puede destacar a la hora de hablar de los videojuegos es el enorme éxito que tienen entre niños, jóvenes y adultos, así como los motivos de dicho éxito.

Se pueden destacar dos de estos motivos. El primero hace referencia a la similitud de los valores, actitudes y comportamientos que promueven los videojuegos y los dominantes en la sociedad actual. Y en segundo lugar, destaca que los videojuegos incluyen muchos de los requisitos que debe tener una buena enseñanza.

Afinidad con los valores dominantes

Como se introdujo anteriormente, muchos de los valores dominantes en nuestra sociedad actual aparecen en los videojuegos con mayor éxito y aceptación, valores como el sexismo, la competición, el consumismo, la velocidad, la violencia, la agresividad, etc. Queda claro que no tienen porqué ser valores correctos desde el punto de vista moral o ético, pero sin duda suponen un gran estímulo para el jugador, lo que convierte a los

videojuegos en un medio de entretenimiento con tanto éxito.

Entre los valores promovidos por los videojuegos y que despiertan un interés especial se pueden destacar los siguientes **[3]**:

- La competitividad. Supone uno de los motores de nuestra sociedad, presente en muchos aspectos de ésta, como el trabajo, el deporte, la familia, etc. Y por supuesto desempeña un papel clave en los videojuegos, ya que supone un aspecto de motivación y superación, tanto si se trata de competitividad con otros como si es con uno mismo.

- La violencia. Presente en gran parte de los videojuegos actuales y sin duda, uno de los temas con mayor éxito entre éstos. Es una realidad, que por desgracia se encuentra presente en el día a día de nuestra sociedad, las guerras o la violencia de género son sólo algunas formas de violencia presente en telediarios, películas, etc.

- Sexismo y erotismo. Utilizado como recurso visual y de atracción que crea un sentimiento positivo, de manera que se usa mucho en publicidad generando dicha percecpción positiva sobre el producto anunciado. De esta misma forma se usa en los videojuegos.

- La velocidad. Otro de los géneros de éxito entre los videojuegos son los de conducción, de motos, coches o de cualquier otra cosa. De alguna manera satisfacen una de las necesidades de la sociedad moderna, que es la sensación de correr más que nadie, de arriesgar y de sentir la adrenalina de la velocidad.

Psicología del aprendizaje y Videojuegos

Resulta muy interesante analizar los videojuegos desde el punto de vista de las teorías educativas y motivacionales. Y si tomamos como referencia la "Teoría del aprendizaje socia", observamos en los videojuegos muchos aspectos en común con dicha teoría, como pueden ser **[3]**:

- El carácter lúdico de los aprendizajes.
- El ritmo individual de cada participante.
- Los aplausos, los gritos del público.
- El conocimiento inmediato de los resultados.
- El conocimiento claro de las tareas y objetivos a conseguir.
- La posibilidad de repetir y corregir el ejercicio.
- La recompensa inmediata después de cada logro.
- La dificultad creciente y progresiva de las habilidades.
- La constante superación del propio nivel.
- La estimulación visual y auditiva de los juegos.

En este ámbito, destaca un nombre propio, James Paul Gee, cuyo libro *"Lo que nos enseñan los videojugos sobre el aprendizaje y el alfabetismo"* rompe con muchas de las ideas estereotipadas que tienen muchos adultos sobre videojuegos. Gee afirma que los juegos no son solamente una fuente de entretenimiento, sino también un modelo a seguir sobre cómo deberían funcionar las escuelas, en sí, sobre cómo debería encararse todo proceso de aprendizaje. En este libro, Gee desarrolla una lista de 36 principios que, según él, los procesos de aprendizaje pueden aprovechar de los videojuegos. Del libro se extraen una serie de mecanismos usados por los videojuegos y aplicables a la enseñanza, como pueden ser **[4]**:

- Jugar o aprender, en la frontera de nuestras posibilidades puede ser altamente motivador. Los videojuegos buscan un nivel alto de dificultad, pero nunca llegan a ser imposibles, de forma que el jugador se verá motivado por el reto y hará uso de toda su capacidad para mejorar y superar el nivel.

- Los videojuegos potencian el aprendizaje incremental. En la mayoría de los videojuegos los primeros niveles o pantallas son más sencillos que los niveles posteriores. De hecho en muchos de ellos, los primeros niveles se usan a modo de tutoriales, para que el jugador aprenda una serie de habilidades que necesitará usar posteriormente.

- Además los videojuegos usan ciclos de pericia. De esta forma el jugador una vez que se ha enfrentado a un reto o ha aprendido una habilidad, se sigue enfrentando al mismo en niveles posteriores para así alcanzar un cierto control sobre la situación y posteriormente una vez adquirido, se le plantearán nuevos retos, repitiéndose así el proceso de aprendizaje.

Por otro lado, la competitividad es un mecanismo utilizado por muchos videojuegos para mantener el interés del jugador y resulta clave de forma indirecta en la educación del mismo. Existen muchas formas de competición, bien sea con uno mismo, contra compañeros o amigos, o contra gente que ni conoces, pero que tiene un record de puntuación superior al propio. El éxito en esta competición sólo se consigue con la repetición, dedicando horas a perfeccionar nuestras habilidades, de manera que el jugador en su búsqueda del triunfo no para de aprender. Además, otro método muy utilizado por los videojuegos para mantener el interés del jugador es un sistema de recompensas. A todos nos gusta que nos valoren nuestro esfuerzo y nuestros logros en nuestra vida diaria, y el caso de los videojuegos no es distinto, por lo que ver recompensados los avances estimulan al jugador a seguir jugando, mejorando y de esta forma aprendiendo.

Es en estos aspectos en los que se basa el videojuego desarrollado para la realización del presente trabajo de fin de grado. En primer lugar usa un mecanismo de puntuaciones, para mantener la competitividad con uno mismo y con el resto de compañeros de clase. Y por otro lado, al tratarse de una actividad propuesta por el profesor existe, a modo de recompensa, la satisfacción de saber que nuestro profesor está al tanto de nuestros avances y de que nos tomamos en serio esta actividad perteneciente a la correspondiente asignatura. Además, el mecanismo establecido para recuperar vidas y poder seguir jugando se basa en el acierto de una serie de preguntas pertenecientes al contenido de la asignatura, de manera que el alumno cuanto más juega, incentivado por la competición, más aprende y a la vez más satisfecho se siente de que sus avances en el juego, así como sus respuestas correctas sean conocidas por el profesor.

2.1.3. Algunos juegos educativos con éxito

En este apartado se van a exponer brevemente algunos de los videojuegos que han tenido mayor éxito a lo largo de los años y que tienen un claro fin educativo.

En primer lugar, cabe destacar que la empresa NINTENDO se ha convertido en el líder de esta nueva tendencia, desarrollando una amplia gama de videojuegos que activan la mente, aumentan el vocabulario e incluso sirven para aprender un nuevo idioma, a la vez que el jugador se divierte. La conocida consola de pantalla táctil, la Nintendo DS (NDS) es con diferencia la que más juegos educativos tiene, por lo que se ha ganado tanto por méritos propios como por falta de competencia el título de mejor consola educativa.

Entre los juegos educativos de mayor éxito desarrollados por Nintendo se encuentran los siguientes:

Brain Training del Dr. Kawashima

Se trata de un videojuego de lógica y puzles, que fomenta la capacidad cerebral de sus usuarios. Su estructura se basa en una serie de ejercicios cuya finalidad es ejercitar la mente. Las actividades que engloba son de diverso carácter, desde ejercicios de cálculo, retención de datos, ejercicios de lectura, ortografía hasta juegos de agilidad mental, como sudokus.

En este juego se reta al jugador con 17 ejercicios: desde el clásico piedra, papel y tijera, un juego para crear palabras, otro en el que hay que dar el cambio correcto en rápidas transacciones monetarias o una prueba en la que el usuario debe seguir una partitura musical tocando un piano virtual en la pantalla táctil de Nintendo DS **[5]**.

En la **Figura 2.2** se muestra una captura del videojuego «Brain Training» para Nintendo DS.

Figura 2.2: Videojuego «Brain Training» para NDS
[Imagen tomada de la página www.meristation.com]

Training for your eyes

Se trata de un videojuego, avalado por el prestigioso doctor Hisao Ishigaki, una autoridad en entrenamiento visual para deportistas de élite, que ayuda a reforzar cinco habilidades visuales distintas.

Con Training for Your Eyes el jugador podrá entrenar su vista a través de diferentes ejercicios basados en el uso de la visión dinámica, de la visión momentánea, del movimiento de los ojos, de la visión periférica, de la coordinación ojo-mano y mediante ejercicios para relajar la vista **[5]**.

En la **Figura 2.3** se muestra una captura del videojuego "Training for your eyes" para Nintendo DS.

Figura 2.3: Videojuego "Training for your eyes" para NDS
[Imagen tomada de la página www.vandal.net]

Por otro lado, la empresa SONY también dispone de algunos videojuegos educativos, aunque en menor cantidad que la marca Nintendo. Entre éstos destaca por ejemplo:

Buzz! el Mega Concurso

Se trata de una simulación de los típicos concursos televisivos de pregunta y respuesta sobre cultura general y que tanto éxito han tenido y tienen.

Este videojuego supone la continuación de una saga, incluyendo mejoras a su predecesor, «Buzz El Gran Reto», de manera que incluye hasta un total de 5.000 nuevas preguntas de conocimiento general, televisión, películas, música, deportes, ciencia, naturaleza y muchos más temas. Además incluye gran variedad de imágenes y vídeos de películas, deportes, naturaleza y televisión.

Por último, permite elegir entre el modo fácil, normal o difícil para adaptarse a las característas y a la edad de los jugadores y también permite ajustar la duración del juego según las necesidades del grupo **[6]**.

En la **Figura 2.4** se muestra una captura del videojuego «Buzz el Mega Concurso»

para Sony.

Figura 2.4: Videojuego «Buzz el Mega Concurso» para Sony
[Imagen tomada de la página www.viciojuegos.com]

Aprende con Pipo

Por último merece ser nombrada la colección de videojuegos llamada "Aprende con Pipo". Se trata de una serie de videojuegos para los más pequeños, de entre 4 y 10 años a través de los cuales los niños pueden aprender sobre diferentes temas a la vez que se divierten jugando. Cada título está enfocado para que los más pequeños de la casa aprendan sobre temas diferentes como vocabulario, matemáticas, aprender a leer, geografía, música, etc. Algunos de los títulos de esta colección son **[7]**:

- Aprende a leer con Pipo.
- Aprende Inglés con Pipo.
- Aprende Música con Pipo.
- Geografía con Pipo.
- El Cuerpo Humano con Pipo.
- etc.

En la **Figura 2.5** se muestra una captura del videojuego "Aprende con Pipo".

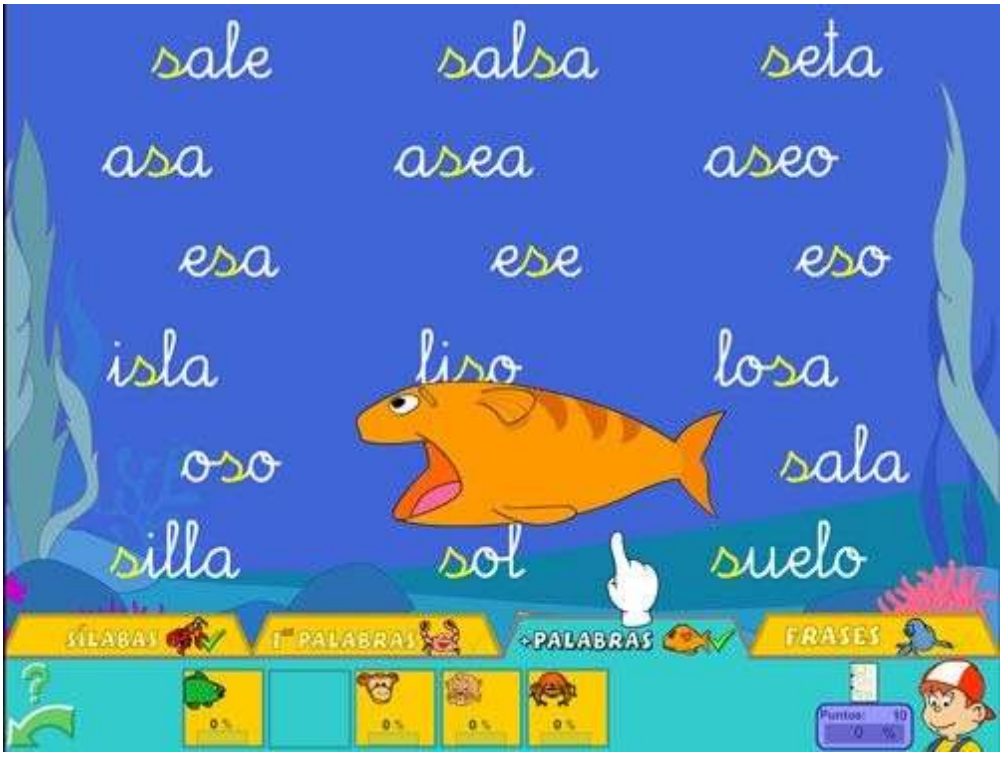

Figura 2.5: Videojuego «Aprende con Pipo»
[Imagen tomada de la página www.identi.li]

2.1.4. Videojuegos en la actualidad

La venta de videojuegos y consolas es en la actualidad la industria del entretenimiento que más dinero mueve, por encima de los tradicionales líderes del sector, la industria cinematográfica y la musical. No es, por tanto, de extrañar que la profesión de programador de videojuegos sea hoy de las preferidas entre los ingenieros de informática y telecomunicaciones. Sin embargo, en nuestro país la situación no es tan halagüeña ya que no existe una industria tan poderosa que demande profesionales como en otros paises. Hoy por hoy España es importadora de videojuegos y exportadora de programadores, ya que aunque no son muchos, los españoles están entre los más cotizados [8].

La industria del videojuego se estructura en dos grandes campos: las videoconsolas y los ordenadores. La diferencia fundamental radica en que las primeras están diseñadas casi exclusivamente para ejecutar videojuegos, mientras que los segundos se extienden hacia otros ámbitos.

Las empresas importantes en cuanto a videoconsolas son sólo tres. Las japonesas Nintendo (con Wii y NDS) y Sony (PlayStation y PSP) y la norteamericana Microsoft, con Xbox [8].

En la actualidad estas tres empresas parecen muy asentadas y se reparten el mercado equitativamente, lo que hace presagiar que ninguna otra compañía entrará en juego en los próximos tiempos. Sin embargo, esta división no ha sido siempre así. Hasta hace unos años había que contar con otro gigante: Sega; una empresa que debido a pérdidas económicas y a reestruturaciones internas se vió obligada a dejar de fabricar sus consolas. Otro ejemplo más reciente lo encontramos en la finlandesa Nokia, que

lanzó en 2003 la N-Gage, un híbrido entre móvil y consola que no llegó a cuajar en el mercado.

En la **Figura 2.6** se representa el volúmen de ventas de videoconsolas a nivel mundial en el año 2010.

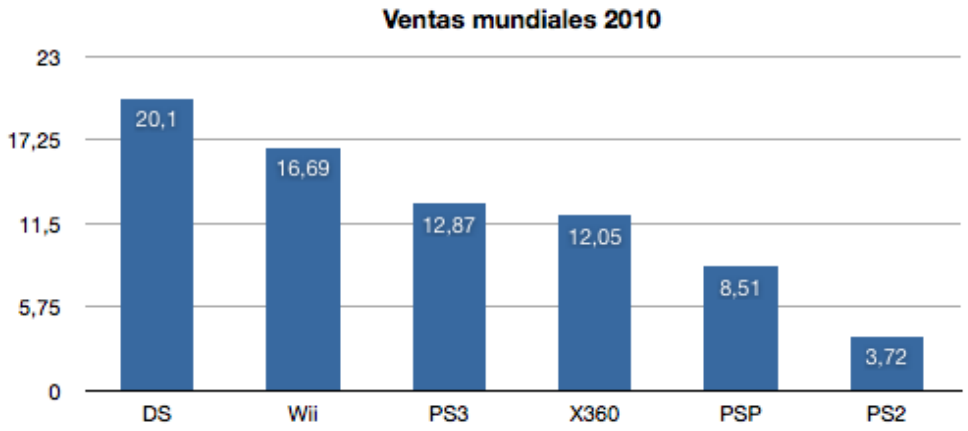

Figura 2.6: Ventas mundiales 2010
[Imagen tomada de la página www.vidaextra.com]

En el caso de los videojuegos en ordenador es bien distinto. En primer lugar porque casi todos ellos trabajan bajo un mismo sistema operativo: Windows; y en segundo lugar porque debido a la gran cantidad de configuraciones que puede tener un PC, ninguna empresa tiene el monopolio sobre un sistema concreto.

En los últimos tiemos han surgido algunas voces importantes de la industria para decir que el PC como plataforma de juegos está destinado a morir; la principal razón que abogan es la elevadísima piratería que tiene.

Correspondencia entre videojuegos y cine

No es ninguna novedad decir que a día de hoy la industria del cine se encuentra estancada. Esto es debido, entre otras cosas, a la falta de nuevas ideas. La carencia se observa todavía mejor si vemos que muchos directores están recurriendo a los propios videojuegos para subsanar su escasez de originalidad.

Los mundos y personajes que ofrecen los juegos parecen ser suficientes para lograr datos positivos de audiencia. Tan solo hay que ver como el personaje de Lara Croft recaudó 48 millones de dólares el primer fin de semana con una película que poco o nada tenía que ver con el videojuego.

Fijándonos en aspectos más profundos, las nuevas tecnologías permiten a los diseñadores de juegos lograr escenas de gran calidad y gran realismo. Tanto que en ocasiones algunas películas beben directamente de ellas, copiando no solo cada plano, si no también la estructura narrativa que tiene el juego. El ejemplo más representativo lo encontramos en «Doom», una película del año 2005 basada en el juego con el mismo nombre. En la misma vemos como en sus últimos minutos la cámara se mueve justo detrás del arma del protagonista y todas las imágenes pasan a ser generadas por ordenador.

El caso contrario, es decir, el paso del cine a los videojuegos, también se produce con mucha frecuencia. Generalmente se aprovecha para lanzar el videojuego bajo la

misma campaña de marketing de la película. Las ventas fáciles son el objetivo de esta serie de productos que en casi todos los casos tienen una calidad bastante baja [8].

Futuro

La industria del videojuego no para de crecer desde que nació, y no hay presagios de que pare en un futuro próximo. Además es muy probable que se convierta en el principal medio de entretenimiento de las próximas generaciones.

Todos los videojuegos se encaminan hacia el fotorrealismo, y debido a ello seguramente triunfe quien sea capaz de innovar lejos del aspecto técnico. El gasto en desarrollo seguirá incrementándose como hasta ahora y la publicidad será un buen inyector económico. Además todo parece indicar que la distribución por medios físicos va a desaparecer, en su lugar se bajarán de Internet.

En conclusión, todo parece indicar que los videojuegos seguirán en expansión. Las audiencias aumentan y muchas empresas verán en ellos un filón comercial [9].

2.2. Herramientas para el desarrollo

Por otro lado, una de las primeras decisiones que se debe tomar para la realización del presente proyecto es el tipo de videojuego que se quiere realizar. De esta decisión depende el motor de videojuegos que se utilizará para desarrollarlo, así como el lenguaje de programación elegido para tal fin.

2.2.1. Motores de videojuegos

A continuación se exponen varias opciones tenidas en cuenta de entre la gran cantidad de motores de videojuegos existentes en la actualidad.

2.2.1.1. Torque 3D

El motor Torque Game Engine, o TGE, es un Motor de videojuego 3D inicialmente desarrollado por Dynamix. Esta versión inicial fue desarrollándose con el tiempo y dando lugar a diferentes motores derivados. Uno de éstos, Torque 3D, constituye el motor insignia actual desarrollado por GarageGames. Torque 3D fue lanzado bajo la licencia MIT[1] el 20 de septiembre de 2012 y se encuentra actualmente en la versión 3 [10].

En la **Figura 2.7** se muestra un ejemplo del programa editor de Torque 3D.

[1]Es una de tantas licencias de software empleadas por el Instituto Tecnológico de Massachusetts, bajo la cual se permite reutilizar el software así licenciado tanto para ser software libre como para ser software no libre.

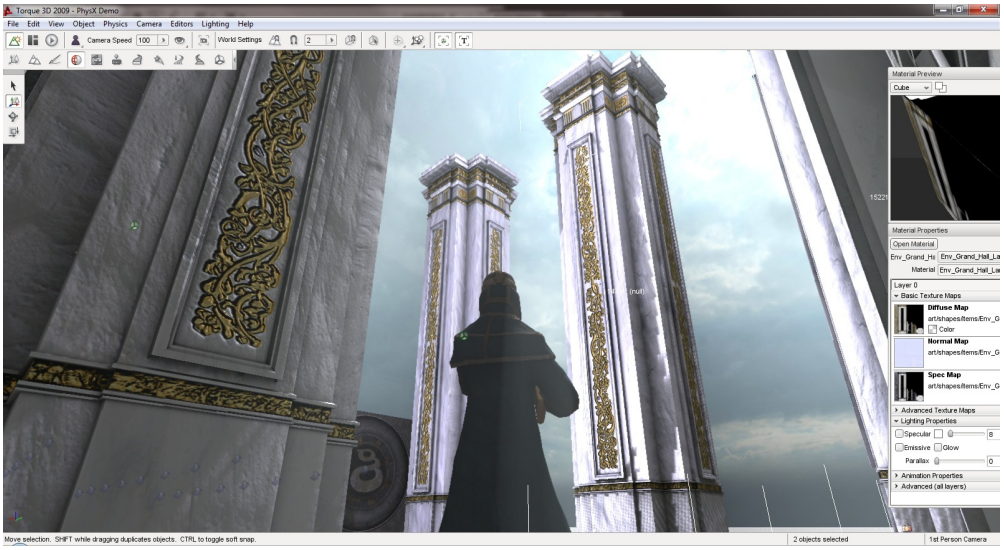

Figura 2.7: Editor de Torque 3D
[Imagen tomada de la página www.eat3d.com]

Torque 3D es una herramienta de creación de videojuegos que nos permitirá pasar por prácticamente todas las etapas de desarrollo de un juego. Creando cada elemento de nuestra propia aventura, nuestro mundo, con sus diferentes texturas, personajes, iluminación, etc. e incluso proporciona ayuda para publicarlo en internet, en plataformas como Kongregate, Facebook o BigFish.

2.2.1.2. Unity

Se trata de un motor de videojuego multiplataforma creado por Unity Technologies.

Unity está disponible como plataforma de desarrollo para Windows y OS X, y permite crear juegos para Windows, OS X, Linux, Xbox 360, PlayStation 3, PlayStation Vita, Wii, iPad, iPhone, Android y Windows Phone. Su última versión, la 4.3, fue lanzada el 12 de noviembre de 2013. Desde la página oficial se pueden descargar dos versiones: Unity y Unity Pro, siendo esta última de pago y ofreciendo características adicionales **[11]**.

Unity supone un poderoso motor de renderizado totalmente integrado con un conjunto completo de herramientas intuitivas y flujos de trabajo rápido para crear contenido 3D interactivo, además de ofrecer herramientas dedicadas también a la creación de contenido 2D.

Incluye un Editor intuitivo y extensible, que constituye el espacio de trabajo donde poder agrupar rápidamente todas las escenas.

Por otro lado, es importante hablar de Mecanim. Se trata de la tecnología de animación de Unity, poderoso, flexible y con características únicas, que permite dar vida a cualquier personaje u objeto con un movimiento increíblemente natural y fluido. Esta tecnología ha estado en desarrollo durante años, primero por la empresa que llevaba el mismo nombre y posteriormente por las oficinas de Unity en Canadá **[12]**.

En la **Figura 2.8** se muestra un ejemplo del Editor de Unity 3D.

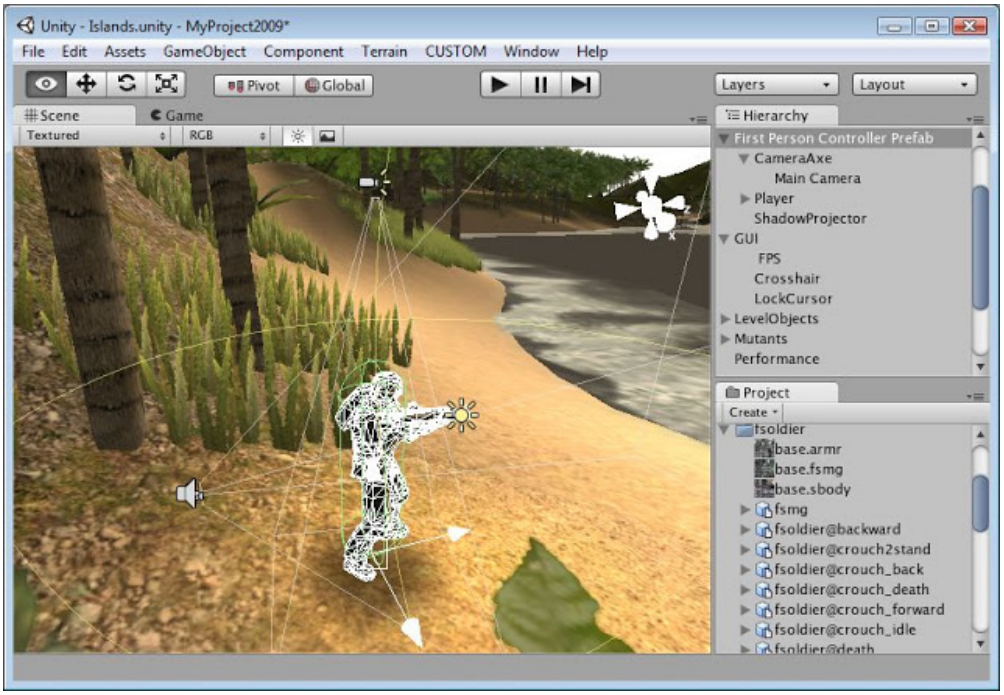

Figura 2.8: Editor de Unity 3D
[Imagen tomada de la página www.inatel.br]

2.2.1.3. Pygame

Pygame es un conjunto de módulos del lenguaje Python que permiten la creación de videojuegos en dos dimensiones de una manera sencilla. Está orientado al manejo de sprites[2]. Pygame incluye gráficos y bibliotecas de sonido diseñados para ser utilizados directamente bajo Python. Funciona como interfaz de las bibliotecas SDL (Simple DirectMedia Layer), que a su vez son un conjunto de bibliotecas desarrolladas en lenguaje C, proporcionando las funciones básicas para realizar operaciones de dibujado 2D, gestión de efectos de sonido y música, y carga y gestión de imágenes. Pygame es altamente portable y funciona en casi todas las plataformas y sistemas operativos **[14]**.

En la **Figura 2.9** se muestra un ejemplo de la pantalla de un videojuego creado con la librería Pygame.

[2]Un sprite es cualquier objeto que aparece en nuestro videojuego. Normalmente tiene asociado algunos atributos, siendo los más básicos una imagen y una posición, que se almacenan en una estructura o clase. Pueden ser estáticos o dinámicos **[13]**.

Figura 2.9: Ejemplo con Pygame
[Imagen tomada de la página www.blendedtechnologies.com]

2.2.2. Lenguajes de programación y entornos de desarrollo

A continuación se muestran varias opciones para elegir un lenguaje de programación y un entorno de desarrollo apropiado para dicho lenguaje.

2.2.2.1. Java y Eclipse

Java

El lenguaje de programación Java fue originalmente desarrollado por James Gosling de Sun Microsystems (la cual fue adquirida por la compañía Oracle) y publicado en 1995 como un componente fundamental de la plataforma Java de Sun Microsystems. Su sintaxis deriva en gran medida de C y C++ pero tiene menos utilidades de bajo nivel que cualquiera de ellos. Es un lenguaje de programación orientado a objetos y basado en clases.

Una característica importante de este lenguaje es su intención de permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo, lo que quiere decir que el código que es ejecutado en una plataforma no tiene que ser recompilado para ejecutarse en otra.

Además cabe destacar, que la memoria se gestiona por medio de un recolector de basura, éste se trata de un mecanismo implícito que reserva, libera y compacta espacios de memoria **[15]**.

Eclipse

Eclipse es un programa informático compuesto por un conjunto de herramientas de programación de código abierto multiplataforma. Se trata de una plataforma de desarrollo, diseñada para ser extendida de forma indefinida a través de plug-ins. No fue creado pensando en un único lenguaje de programación, sino que es un IDE (Integrated Development Environment) genérico, aunque es típicamente usado para el lenguaje Java usando el plug-in JDT. Pero dado que es un IDE genérico permite su adaptación a otros lenguajes, como Python, mediante una simple configuración, que consiste en la instalación del módulo PyDev en el IDE **[16]**.

En la **Figura 2.10** se muestra una vista del IDE Eclipse configurado para trabajar con el lenguaje de programación Python.

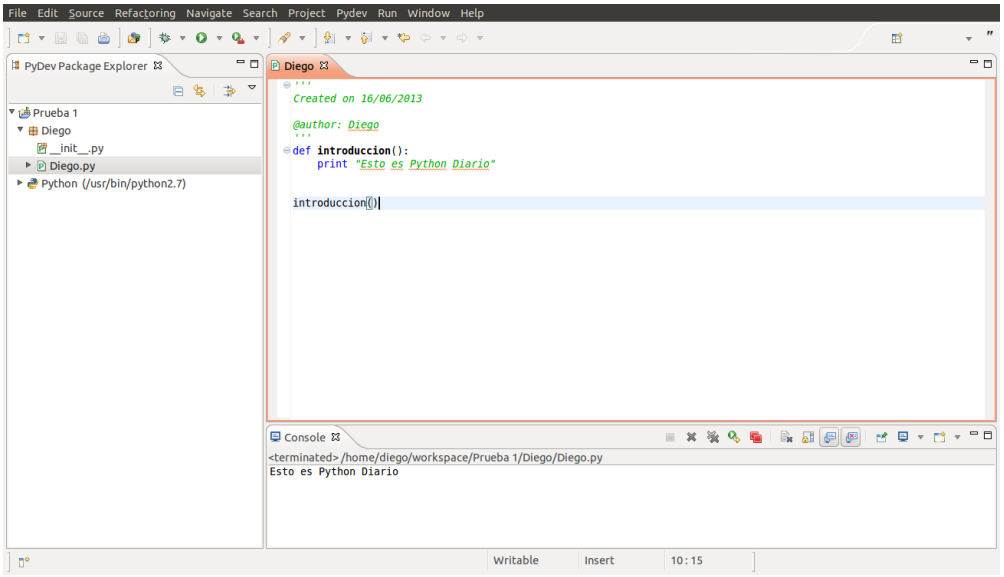

Figura 2.10: Eclipse para python
[Captura de pantalla del propio programa]

2.2.2.2. C# y Microsoft Visual Studio

C#

Se trata de un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET[3], que después fue aprobado como un estándar por la ECMA e ISO. C# es uno de los lenguajes de programación diseñados para la infraestructura de lenguaje común[4].

Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET, similar al de Java, aunque incluye mejoras derivadas de otros lenguajes.

Aunque C# forma parte de la plataforma .NET, ésta es una API (Application Pro-

[3]La plataforma .NET es un componente de software que puede ser añadido al sistema operativo Windows. Provee un extenso conjunto de soluciones predefinidas para necesidades generales de la programación de aplicaciones, y administra la ejecución de los programas escritos específicamente con la plataforma.

[4]La infraestructura de lenguaje común es una especificación estandarizada que describe un entorno virtual para la ejecución de aplicaciones, cuya principal característica es la de permitir que aplicaciones escritas en distintos lenguajes de alto nivel puedan luego ejecutarse en múltiples plataformas tanto de hardware como de software sin necesidad de reescribir o recompilar su código fuente.

gramming Interface)[5], mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. El entorno de desarrollo por excelencia para el lenguaje C# es Microsoft Visual Studio **[17]**.

Microsoft Visual Studio

Es un entorno de desarrollo integrado (IDE) para sistemas operativos Windows. Soporta múltiples lenguajes de programación tales como C++, C#, Visual Basic .NET, Java, Python, etc.

Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET **[18]**.

En la **Figura 2.11** se muestra la pantalla de trabajo del entorno de desarrollo Microsoft Visual Studio.

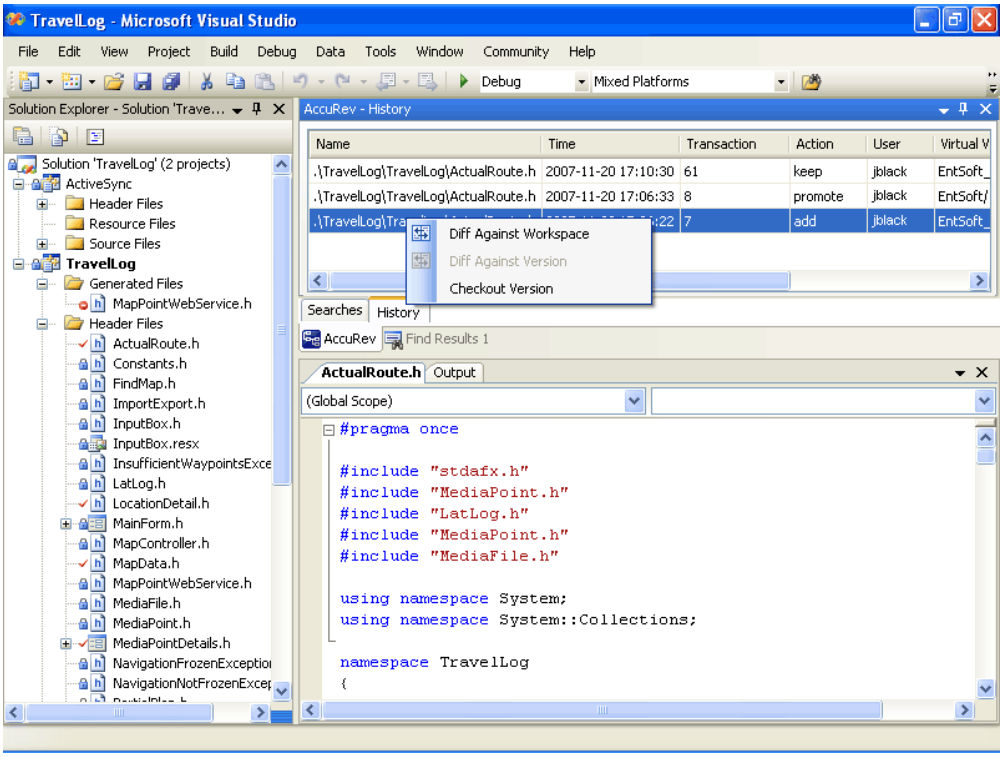

Figura 2.11: IDE Microsoft Visual Studio
[Imagen tomada de la página www.visualstudiogallery.msdn.microsoft.com]

2.2.2.3. Python y Eclipse

Python

Python es un lenguaje de programación interpretado[6] cuya filosofía hace hincapié en una sintaxis que favorezca un código legible.

El intérprete de Python "compila" a bytecode el programa, lenguaje máquina optimizado para una máquina virtual de Python. La primera vez que ejecutamos el pro-

[5]Es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

[6]Un lenguaje interpretado es un lenguaje de programación que está diseñado para ser ejecutado por medio de un intérprete, en contraste con los lenguajes compilados.

grama éste compila a bytecode para la máquina virtual de Python. Después se ejecuta el bytecode optimizado generado, así se ejecuta más rápido **[19]**.

Python es un lenguaje de programación multiparadigma. Esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos: programación orientada a objetos, programación imperativa y programación funcional. Otros paradigmas están soportados mediante el uso de extensiones.

Entre las características de Python destaca que usa tipado dinámico[7] y conteo de referencias[8] para la administración de memoria **[19]**.

Por último, considero interesante destacar el gran calado que ha tenido entre los programadores lo que se conoce como **Filosofía Python**. Entre ellos se refieren como "pythonico" a aquel código que sigue los principios de Python de legibilidad y transparencia, mientras que el código opaco y ofuscado se cataloga como «no pythonico». Estos principios fueron descritos por el desarrollador de Python Tim Peters en *El Zen de Python* **[20]** a través de los cuales se transmite el tipo de lenguaje que es Python, donde se busca lo bello, lo sencillo, lo explícito y lo simple entre otras cosas **[19]**.

En cuanto al programa Eclipse, se trata de un entorno de desarrollo del que ya hemos hablado anteriormente en el apartado **2.2.2.1** junto con el lenguaje de programación Java. Por lo tanto, en este espacio lo único que cabe añadir es que dado que se trata de un IDE genérico se puede adaptar mediante una simple configuración para ser utilizado junto con el lenguaje Python. Esta configuración consiste en la instalación del módulo PyDev en el IDE Eclipse.

2.2.3. Bases de datos

Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

El término de bases de datos fue escuchado por primera vez en 1963 en un simposio[9] celebrado en California, USA.

Desde el punto de vista informático, la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.

Según la variabilidad de la base de datos podemos diferenciarlas en dos tipos **[21]**:

- Bases de datos estáticas: son bases de datos de solo lectura.
- Bases de datos dinámicas: son bases de datos donde la información almacenada se modifica con el tiempo, permitiendo operaciones como actualización, borrado y edición de datos, además de las operaciones fundamentales de consulta.

[7] Un lenguaje de programación es dinámicamente tipado si una misma variable puede tomar valores de distinto tipo en distintos momentos.

[8] Conteo de referencias es una técnica para contabilizar las veces que un determinado recurso está siendo referido. Por lo general ese recurso son bloques de memoria y la técnica permite establecer cuando no existe ninguna referencia a ese bloque y éste puede ser liberado.

[9] Un simposio es una reunión de especialistas en una materia para tratar y discutir sobre algo concreto relacionado con su especialidad.

También se puede hacer una clasificación de las bases de datos atendiendo a su modelo de administración de datos, de esta forma se diferencian las bases de datos jerárquicas, las bases de datos de red, las bases de datos multidimensionales, las bases de datos orientadas a objetos, las bases de datos documentales, las bases de datos deductivas y por último las bases de datos relacionales.

Entrando más en detalle en la descripción de las bases de datos relacionales, decir que son el modelo utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. La idea de estas bases de datos es el uso de "relaciones", entendidas como conjuntos de datos llamados "tuplas". Cada relación constituye una tabla que a su vez está compuesta por registros (las filas de una tabla), que representarían las tuplas, y campos (las columnas de una tabla).

El lenguaje más habitual para construir las consultas a bases de datos relacionales es SQL, *Structured Query Language* o *Lenguaje Estructurado de Consultas*, un estándar implementado por los principales motores o sistemas de gestión de bases de datos relacionales **[22]**.

También es importante mencionar a las bases de datos no relacionales, comúnmente conocidas como NoSQL. Este concepto hace referencia a una amplia clase de sistemas de gestión de bases de datos que difieren del modelo clásico del sistema de gestión de bases de datos relacionales en aspectos importantes, entre los que destaca que no usan SQL como el principal lenguaje de consultas. En este tipo de bases de datos, los datos almacenados no requieren estructuras fijas como tablas.

A continuación, se presentan algunos de los sistemas de gestión de bases de datos que se han considerado previamente a la realización de la aplicación.

2.2.3.1. MongoDB

MongoDB (de la palabra en inglés "homongous"que significa enorme) es un sistema de base de datos NoSQL orientado a documentos, desarrollado bajo el concepto de código abierto[10].

MongoDB forma parte de la nueva familia de sistemas de base de datos NoSQL. En vez de guardar los datos en tablas como se hace en las bases de datos relacionales, MongoDB guarda estructuras de datos en documentos tipo JSON[11] con un esquema dinámico (MongoDB llama ese formato BSON), haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida **[23]**.

El desarrollo de MonogDB empezó en octubre de 2007 por la compañía de software 10gen.

2.2.3.2. SQLite

SQLite es un sistema de gestión de bases de datos relacional, contenida en una relativamente pequeña biblioteca escrita en C. SQLite es un proyecto de dominio público[12]

[10]Código abierto es la expresión con la que se conoce al software distribuido y desarrollado libremente.

[11]JSON, acrónimo de *JavaScript Object Notation*, es un formato ligero para el intercambio de datos.

[12]Por dominio público se entiende la situación en que quedan las obras literarias, artísticas o científicas (lo que incluye programas informáticos) al expirar el plazo de protección del derecho de autor.

creado por D.Richard Hipp.

A diferencia de los sistemas de gestión de bases de datos cliente-servidor, el motor de SQLite no es un proceso independiente con el que el programa principal se comunica. En lugar de eso, la biblioteca SQLite se enlaza con el programa pasando a ser parte integral del mismo. El programa utiliza la funcionalidad de SQLite a través de llamadas simples a subrutinas y funciones. Esto reduce la latencia en el acceso a la base de datos, debido a que las llamadas a funciones son más eficientes que la comunicación entre procesos. El conjunto de la base de datos (definiciones, tablas, índices, y los propios datos), son guardados como un sólo fichero estándar en la máquina host. Este diseño simple se logra bloqueando todo el fichero de base de datos al principio de cada transacción **[24]**.

En su versión 3, SQLite permite bases de datos de hasta 2 Terabytes de tamaño, y también permite la inclusión de campos tipo BLOB[13].

2.2.3.3. MySQL

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones.

MySQL es usado por muchos sitios web grandes y populares, como Wikipedia, Google (aunque no para búsquedas), Facebook, Twitter, Flickr y YouTube.

En cuanto a los lenguajes de programación, destacar que existen varias interfaces de programación de aplicaciones (API) que permiten, a aplicaciones escritas en diversos lenguajes de programación, acceder a las bases de datos MySQL. Lenguajes como por ejemplo C, C++, C#, Pascal, Java (con una implementación nativa del driver de Java), Lisp o Python (cuya interfaz recibe el nombre de MySQLdb).

MySQL es una base de datos muy rápida en la lectura, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de los datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones **[25]**.

2.2.4. Elección final

En este apartado final del presente capítulo, se explica cuales han sido finalmente las herramientas elegidas para llevar a cabo el desarrollo de la aplicación.

2.2.4.1. Python y Eclipse

En primer lugar, se explican los motivos de la elección en cuanto a lenguajes de programación y entornos de desarrollo. Anteriormente se ha nombrado que el lenguaje

Cada legislación nacional contempla un término de años contados desde la muerte del autor, para que estos derechos expiren. Dominio público, en este caso, implica que las obras pueden ser explotadas por cualquier persona, pero siempre respetando los derechos morales de sus autores.

[13]Los BLOB (Binary Large Objects, objetos binarios grandes) son elementos utilizados en las bases de datos para almacenar datos de gran tamaño que cambian de forma dinámica. No todos los Sistemas Gestores de Bases de Datos son compatibles con los BLOB. Generalmente, estos datos son imágenes, archivos de sonido y otros objetos multimedia.

de programación elegido ha sido Python. Esta fue una elección bastante fácil y de hecho fue de las primeras, ya que tenía claro que quería utilizar un lenguaje que no hubiera aprendido anteriormente y que estuviera en auge, es decir que tuviera bastante uso en la actualidad **[26]**. Por otro lado, en cuanto al entorno de desarrollo escogí Eclipse, puesto que ya lo había utilizado anteriormente y estaba familiarizado con su interfaz de trabajo. Además Eclipse se ha ganado fama por ser un buen entorno de desarrollo, con mucha información disponible para resolver cualquier duda surgida.

En la **Figura 2.12** se muestra un gráfico del uso de los principales lenguajes de programación. En ella se puede observar como el lenguaje Python es uno de los más usados en la actualidad.

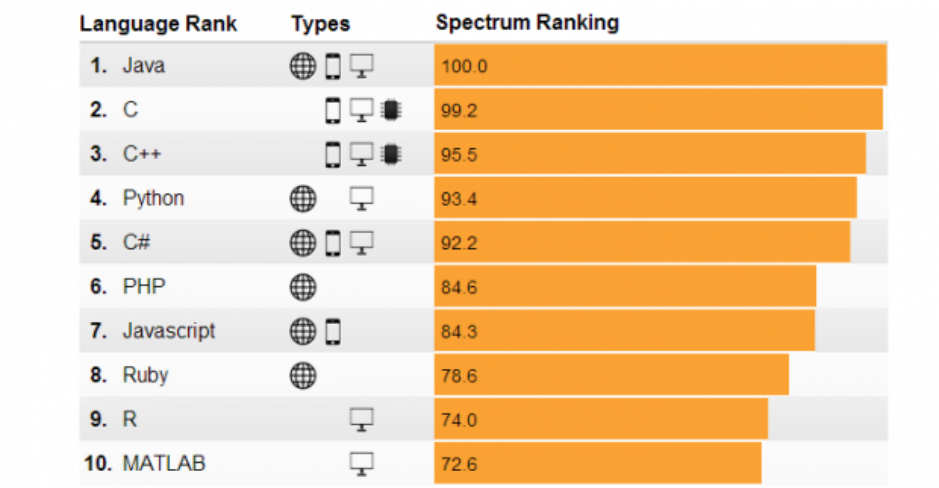

Figura 2.12: IEEE Ranking de los lenguajes de programación.
[Imagen tomada de la página www.dataconomy.com]

2.2.4.2. Pygame

En segundo lugar, se explican los motivos que llevaron a la elección de Pygame como motor para desarrollar el videojuego. Comenzar diciendo, que dicho videojuego debe ser en dos dimensiones y que como se ha explicado anteriormente el lenguaje de programación utilizado ha sido Python. Teniendo en cuenta estos dos hechos y analizando las características de Pygame, se puede observar que la elección de su uso es la más aconsejable, ya que Pygame son un conjunto de módulos propios del lenguaje Python y además se trata de un motor óptimo para desarrollar videojuegos en dos dimensiones. Además, Pygame se trata de una biblioteca muy completa que ofrece gran variedad de recursos y que consta de una amplia y detallada documentación en su página web oficial.

2.2.4.3. SQLite

Por último, toca hablar de los sistemas de gestión de bases de datos. En este sentido, en un primer lugar decidí utilizar MySQL, puesto que se trata de una base de datos relacional de las más prestigiosas, utilizada por la mayoría de los programadores. Por lo tanto, ya que era la primera vez que iba a trabajar con bases de datos, decidí aprender a usar una que es muy utilizada por los profesionales y que además cuenta con amplios

manuales de ayuda. Posteriormente, una vez que la aplicación estaba prácticamente finalizada, decidí cambiar MySQL por SQLite. Esta decisión se basa en la idea de que SQLite posee un módulo llamado SQLite3 que a partir de la versión 2.5 de Python viene integrado con el mismo. Esto supone que no tenemos que realizar instalaciones extra, como en el caso de MySQL, para poder utilizar una base de datos en nuestra aplicación. Además el paso de MySQL a SQLite fue muy sencillo, ya que la sintaxis de ambos es muy similiar, pues ambos utilizan el lenguaje de comandos SQL.

Capítulo 3

Diseño e implementación

En este capítulo se detallará de una forma más técnica en qué consiste la aplicación y cómo se ha implementado la misma. Para ello se explicará la lógica del juego, las posibles jugadas y la interacción del jugador con la aplicación. Además se explicarán los requisitos que debe cumplir, cómo se han gestionado los datos y otros programas utilizados a lo largo del desarrollo del presente trabajo de fin de grado.

3.1. Otros programas utilizados

En esta sección, se presentan algunos de los programas utilizados tanto en el desarrollo de la aplicación como en la redacción de la presente memoria del trabajo de fin de grado.

3.1.1. HeidiSQL

Es un software que consta de una interfaz gráfica muy intuitiva para el manejo y gestión de bases de datos MySQL, tanto locales como remotas. Es conveniente aprender a gestionar la base de datos desde la consola mediante comandos SQL, pero en muchas ocasiones este tipo de programas agilizan el trabajo, ahorrando tiempo al programador [27].

En esta aplicación, HeidiSQL fue utilizada para realizar la base de datos en MySQL. Aunque como ya se ha explicado anteriormente en el apartado 2.2.4, la base de datos fue implementada finalmente utilizando SQLite, por lo que ya no hizo falta el uso de este software.

En la **Figura 3.1** se representa una muestra de la pantalla de trabajo que ofrece el software HeidiSQL para gestionar bases de datos en MySQL.

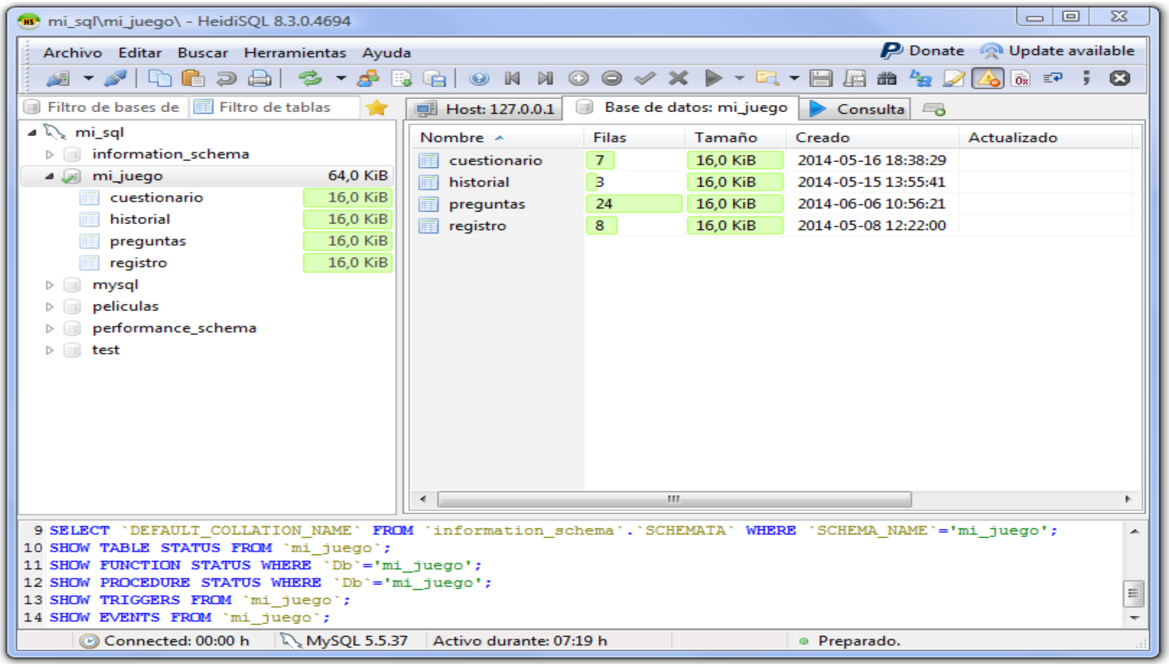

Figura 3.1: Interfaz del software HeidiSQL.
[Captura de pantalla de la propia aplicación]

3.1.2. TeXnicCenter

Se trata de un software libre que sirve como editor de LaTex para Windows. Dispone de diferentes herramientas, lo que le convierten en un programa muy completo. Herramientas como una ventana de compilación, soporte de ayuda y manual de LaTex, entre otras muchas **[28]**.

En la **Figura 3.2** se representa una muestra de la pantalla de trabajo que ofrece el software TeXnicCenter para redactar documentos en LaTex.

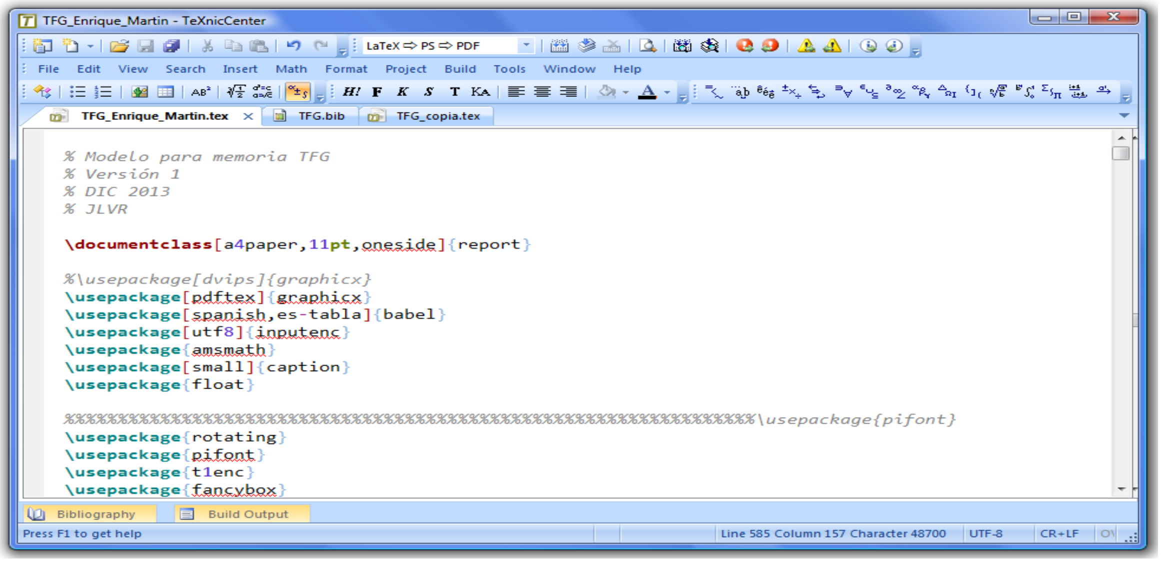

Figura 3.2: Interfaz del software TeXnicCenter.
[Captura de pantalla de la propia aplicación]

3.1.3. Microsoft Visio 2010

Se trata de un software que permite implementar dibujos vectoriales para Microsoft Windows. Permite realizar diagramas de oficinas, diagramas de bases de datos, diagramas de flujo de programas, UML y otros más **[29]**.

En esta aplicación ha sido utilizado para la realización de los diagramas de flujo, incluidos en el apartado **3.6**.

En la **Figura 3.3** se representa una muestra de la pantalla de trabajo que ofrece el software Microsoft Visio 2010 para la elaboración de diagramas de muchos tipos.

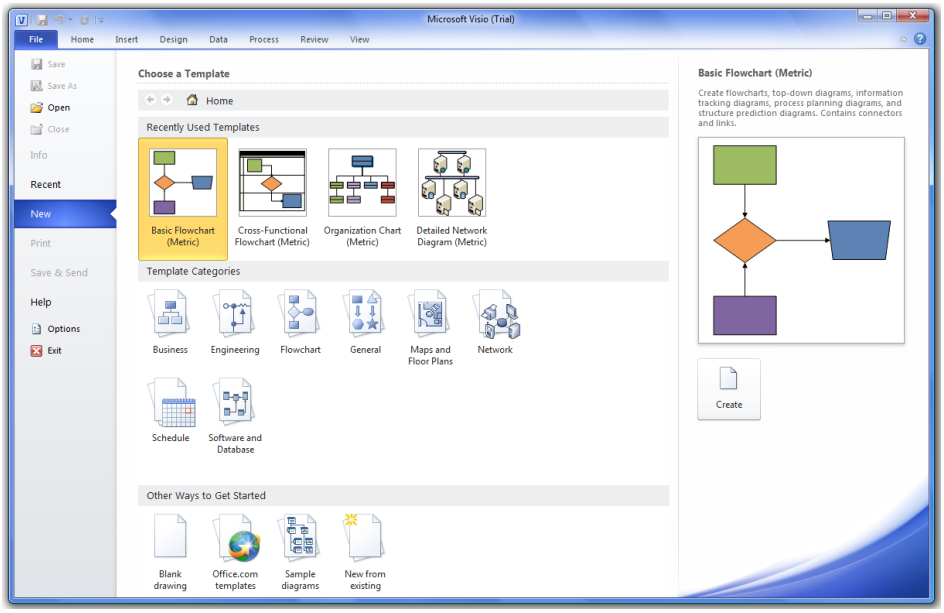

Figura 3.3: Interfaz del software Microsoft Visio 2010.
[Captura de pantalla de la propia aplicación]

3.1.4. PyNSource

Se trata de una herramienta que nos permite realizar diagramas UML a partir de código Python **[30]**.

En esta aplicación ha sido utilizada para la realización de los diagramas UML de las clases, incluidos en el apartado 3.5.

En la **Figura 3.4** se representa una muestra de la pantalla de trabajo que ofrece la herramienta PyNSource para la elaboración de diagramas UML importando un código Python.

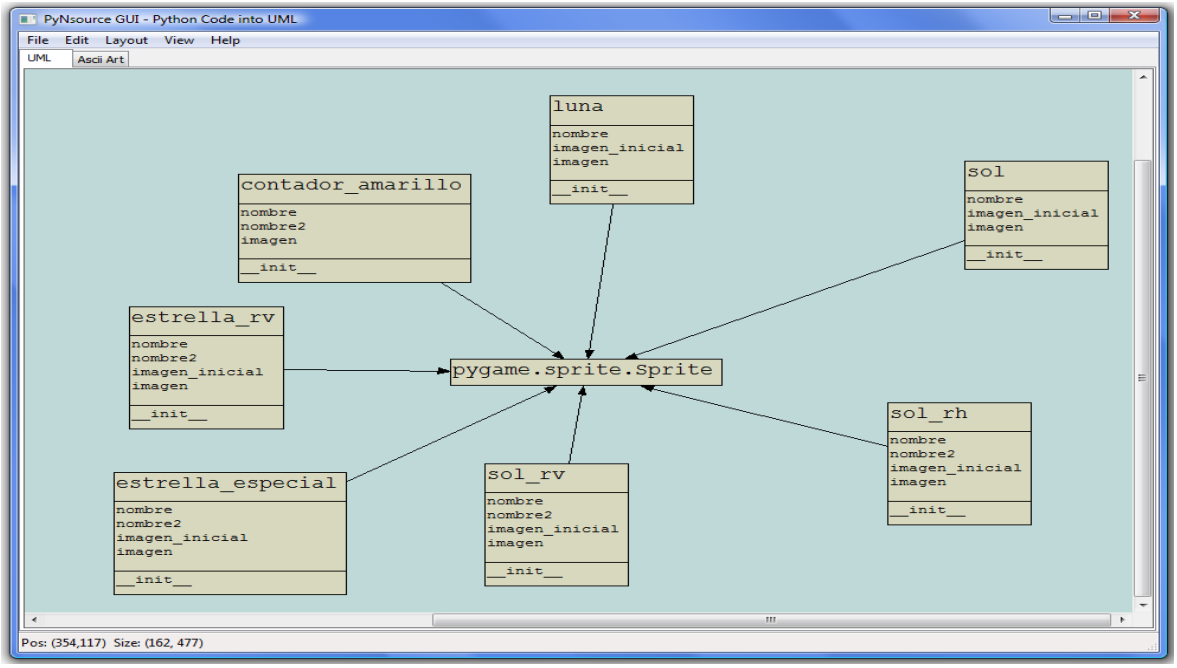

Figura 3.4: Interfaz de la herramienta PyNSource.
[Captura de pantalla de la propia aplicación]

3.1.5. GanttProject

GanttProject se trata de una herramienta muy completa que permite definir de una forma muy visual las diferentes tareas de un proyecto a corto, medio o largo plazo. También permite distribuir las diferentes tareas por personas, de forma que se podrá visualizar claramente el trabajo que debe llevar a cabo cada trabajador. Es tan completa que permite incluso incorporar los días festivos que tiene cada trabajador [31].

En la **Figura 3.5** se muestra la interfaz gráfica que usa GanttProject para realizar la planificación de proyectos.

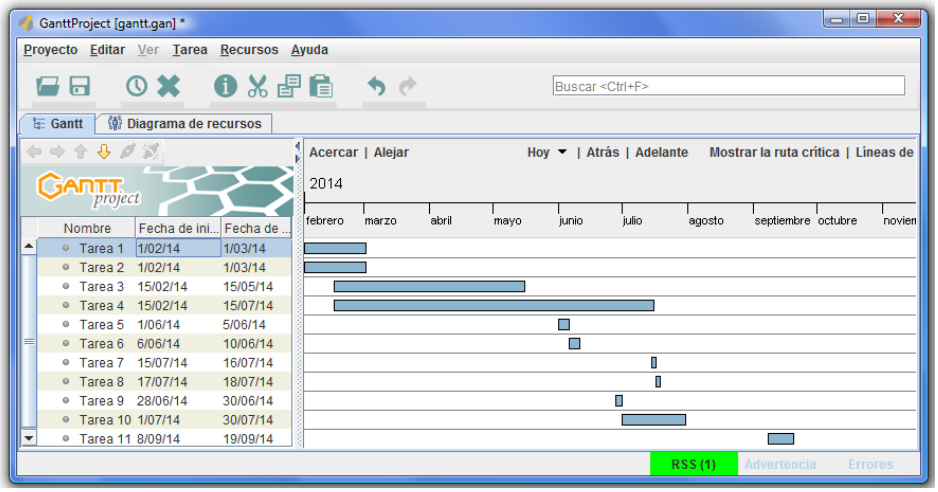

Figura 3.5: Interfaz de la herramienta GanttProject.
[Captura de pantalla de la propia aplicación]

3.2. Dinámica del juego

En esta sección se explicará en qué consiste el juego. Anteriomente se ha mencionado que es un juego del tipo "conecta-tres". Todos los juegos de este tipo tienen una serie de características que los definen. Consisten en un tablero sobre el que se posicionan fichas de diferentes tipos. La idea básica es unir o conectar fichas que sean del mismo tipo.

Por otro lado, decir que la idea del presente juego está basada en el famosísimo Candy Crush Saga, pero con modificaciones.

Añadir también que en este caso, las fichas básicas tienen forma de corazón, sol, estrella y luna, y han sido creadas por el autor del presente trabajo de fin de grado a partir de imágenes cogidas de la web.

A continuación se presentan los movimientos que se le permiten realizar al jugador:

- **Tres iguales:** consiste en unir en línea, tanto vertical como horizontal, tres fichas básicas iguales. Cuando esto ocurre las tres fichas desparecen y caen las fichas que se encuentren por encima suya.

- **Cuatro iguales:** esta jugada consiste en unir las cuatro fichas básicas iguales en una línea, vertical u horizontal. En este caso, surge una ficha con forma de rayo, que tiene dos direcciones. Puede ser vertical, si las cuatro en línea se agruparon horizontalmente o puede ser horizontal si las cuatro fichas se agruparon en una línea vertical.

- **Cinco iguales:** en este caso pueden darse varias posibilidades:

    1. En primer lugar, se pueden agrupar cinco fichas básicas iguales en una línea, tanto horizontal como vertical. Al realizar esta jugada surge otra ficha con forma de bomba.
    2. Por otro lado, las cinco fichas básicas iguales se pueden agrupar formando una "T" o una "L", dando lugar a otra figura con forma de dinamita.

- **Uso del rayo:** esta se trata de una jugada completamente distinta a las posibles en el Candy Crush Saga. El rayo, surgido de la jugada de cuatro en línea explicada anteriormente puede ser usada por el jugador haciendo doble click encima de ella. Cuando esto ocurre desparecerá la fila o la columna donde se encuentre el rayo, en función de la dirección del propio rayo, vertical u horizontal. Al ejecutarse esta jugada, se llevará por delante a toda ficha que encuentre en su camino, haciéndolas desaparecer, excluyendo a unas fichas especiales con forma de «gota», de las cuales hablaremos más adelante.

- **Rayo + rayo:** al ejecutar esta jugada se eliminan la columna y la fila que se cruzan en el punto donde se unieron ambos rayos.

- **Uso de la dinamita:** recordemos que la dinamita surge de agrupar cinco fichas iguales en forma de "T" o "L". Este tipo de ficha puede ser usada por el jugador haciendo doble click encima de ella, al igual que el rayo. Cuando se acciona la dinamita se produce una doble explosión, eliminando en cada una de ellas a sus fichas vecinas.

- **Dinamita + dinamita:** al ejecutar esta jugada se produce, igual que en la jugada anterior, una doble explosión, pero en este caso con el doble de potencia y centrándose en el punto donde se uniron ambas dinamitas. Es decir en cada una de las explosiones desaparecen los vecinos de la ficha central y los vecinos de sus vecinos.

- **Rayo + dinamita:** al ejecutar esta jugada desaparecen tres columnas consecutivas y tres filas consecutivas, siendo la columna y la fila central aquellas que se cruzan en el punto donde se unieron el rayo y la dinamita.

  Recordemos que la bomba surge de alinear cinco fichas básicas iguales, a continuación se presentan las diferentes interacciones entre la bomba y el testo de fichas, dando lugar a diferentes jugadas.

- **Bomba + ficha básica:** esta jugada consiste en unir la bomba con una de las cuatro fichas básicas. Cuando ejecutamos esta jugada el resultado es que desaparecen todas las fichas básicas iguales a la usada en la jugada que haya en el tablero.

- **Bomba + rayo:** al ejecutar esta jugada, un número aleatorio de fichas básicas se convierten en rayo y seguidamente se accionan. Al decir que se accionan me refiero a que es como si hicieramos doble click en cada una de ellas.

- **Bomba + dinamita:** al ejecutar esta jugada dos tipos de fichas básicas elegidas aleatoriamente desaparecen del tablero. Por ejemplo primero desaparecen todos los corazones y una vez reestablecido el tablero, seguidamente desaparecen las estrellas.

- **Bomba + bomba:** al ejecutar esta jugada, desaparecen todas las fichas del tablero.

En todas estas jugadas explicadas se excluyen de su acción a las fichas con forma de "gota", como ya se ha mencionado anteriormente. Estas son fichas que sólo aparecerán en un tipo de nivel y sobre las que no puede actuar el jugador. Es decir, no las podrá mover manualmente, simplemente irán bajando a lo largo del tablero cuando las fichas que se encuentren en su misma columna y por debajo vayan desapareciendo.

También existen otro tipo de fichas, a las que llamaré "contadores", y que pueden ser de los cuatro colores de las fichas básicas. Estos contadores sólo aparecerán también en algunos niveles. Su fúnción es actuar como un contador que decrece con cada jugada que realiza el jugador, de manera que si llegan a cero el jugador perderá. Estas fichas sí se ven afectadas por las jugadas anteriormente explicadas.

3.3. Creación de niveles

En esta sección se explica cómo se crean los diferentes niveles de los que consta el videojuego.

En primer lugar, vamos a diferenciar entre dos tipos de módulos de Python para una mejor comprensión de este apartado. Por un lado, se encuentra el que llamaremos módulo principal y por otro el módulo del nivel.

En el módulo principal es donde se encuentran implementadas las funciones correspondientes a las jugadas permitidas por el juego, que son las explicadas en el apartado 3.2. Es decir, este es el módulo encargado de comprobar si la jugada que intenta realizar el jugador esta o no permitida. También es este módulo principal el encargado de ejecutar la jugada, moviendo las fichas en el tablero como corresponda en cada caso.

Por otro lado, se explican varias características del módulo del nivel:

- Cada nivel esta implementado en un módulo de Python diferente.
- Esto hace que la modificación de las características de cada nivel sea más sencilla.
- También hace que el código esté mejor ordenado, lo que facilita su comprensión.
- Todos los niveles tienen una serie de parámetros comunes, como son:
    1. Movimientos restantes: este parámetro determina de cuantos movimientos dispone el jugador antes de que se acabe la partida.
    2. Puntuación mínima: este parámetro determina uno de los objetivos del nivel. Se trata de la puntuación que debe obtener el jugador para superar ese reto del nivel.
- Por otro lado, los niveles pueden tener otros parámetros, propios de cada nivel y que suponen retos particulares únicamente de dicho nivel. Estos parámetros son:
    1. Combinación de fichas: este parámetro hace referencia a uno de los retos presentes en algunos niveles y que determina el número y el tipo de combinación de fichas que debe realizar el jugador para superar dicho reto.
    2. Gotas: las gotas son un tipo de fichas presentes en algunos niveles. Se trata de fichas que no pueden ser usadas por el jugador. Únicamente se mueven hacia abajo a lo largo del tablero cuando las fichas que están por debajo de ellas desaparecen. Por lo tanto, este parámetro determina el número de gotas que el jugador debe hacer descender a lo largo del tablero para superar este reto.

Por consiguiente, el módulo del nivel se va a encargar se supervisar el valor de los parámetros mencionados anteriormente. De esta forma, comprobará si le quedan o no movimientos al jugador y si ha superado o no los retos propios del nivel en concreto. Así, el módulo del nivel será el encargado de determinar si el jugador obtiene o no la victoria en dicho nivel.

Además, el módulo del nivel realiza otra tarea fundamental:

- Se encarga de registrar cada click que realiza el jugador sobre el tablero.
- Cada jugada requiere de dos clicks. De forma que el módulo del nivel almacena la posición en el tablero de cada dos clicks consecutivos.
- El módulo del nivel, pasa ambas posiciones a cada una de las posibles jugadas, que están implementadas en el módulo principal.

- Será el módulo principal, como ya se mencionó anteriormente, el que determine qué jugada está realizando el jugador, en caso de que sea alguna, a partir de los clicks que le ha proporcionado el módulo del nivel.

Por lo tanto, queda claro que existe una gran interacción entre ambos módulos, el principal y el de nivel, a lo largo de una partida y que son ambos los encargados de gestionar todo lo que ocurre a lo largo de la misma.

Por último, queda decir, que en el módulo del nivel hay otro parámetro fundamental para el correcto funcionamiento de la aplicación. Este parámetro es el de las vidas. El módulo del nivel, antes de ejecutar la partida correspondiente a su propio nivel, debe comprobar el valor de este parámetro. Si no le quedan vidas al jugador, será el módulo del nivel el encargado de ejecutar el módulo correspondiente a las preguntas de test para la recuperación de vidas.

3.4. Requisitos

En primer lugar, se explicarán cuales han sido los requisitos, establecidos a priori, a los que debe satisfacer la aplicación.

Puesto que fueron definidos a priori, previamente a la realización de la aplicación, hay una etapa de diálogo, en este caso entre mi tutora y yo. Este es el momento en el que se debe de dejar claro qué queremos y cómo lo vamos a conseguir. Lo que queremos ya se ha explicado en capítulos anteriores, pero en pocas palabras es un juego que guste al alumno y que a través de él pueda aprender el contenido de la asignatura.

Por otro lado, dentro del cómo lo vamos a hacer, juegan un papel fundamental los requisitos que debe cumplir la aplicación.

A continuación se exponen dichos requisitos:

1. **Registro de usuario:**

   El usuario debe ser capaz de registrarse la primera vez que accede al juego. Deberá introducir el nombre y la contraseña en el espacio reservado para ello. En este momento se creará su perfil de usuario que quedará grabado en la base de datos.

   El nombre de usuario debe ser único, de forma que si el nombre introducido por el jugador ya existe, la aplicación le pedirá que vuelva a intentar el registro. Por el contrario, si el nombre de usuario elegido por el jugador no existiese, la aplicación realizará el registro con éxito.

   Además, en el momento del registro es cuando se cargan las preguntas de test que posteriormente responderá el jugador para recuperar vidas. Pero de esto ya se hablará más tarde, en el apartado correspondiente a la base de datos.

2. **Inicio de sesión:**

   El inicio de sesión tiene lugar cada vez que el jugador inicia la aplicación y siempre y cuando ya haya accedido anteriormente, es decir, el inicio de sesión tendrá lugar si anteriormente dicho usuario ya realizó el registro de usuario. Una vez realizado el inicio de sesión con éxito se deberá cargar el perfil de dicho usuario para

acceder a los datos que la aplicación necesite conocer en cada momento acerca del progreso del usuario en cuestión.

3. **Menú de los niveles:**

   Como todo juego con diferentes pantallas y niveles, debe existir un menú en el que el jugador seleccione el nivel al que quiere jugar. Debe tenerse en cuenta el progreso actual del jugador, de forma que sólo se le permita acceder a los niveles ya superados y al siguiente nivel al último superado.

4. **Ver clasificación:**

   El jugador debe ser capaz de ver sus puntuaciones así como la de los otros jugadores. Esto aumenta la competencia tanto con él mismo como contra los otros jugadores.

5. **Descripción del nivel:**

   Antes de disputar un nivel el jugador debe conocer previamente los retos que debe superar en ese nivel para poder conseguir la victoria.

6. **Recuperar vidas respondiendo preguntas:**

   Uno de los requisitos más importantes de esta aplicación es el que concierne al mecanismo de recuperación de vidas. Como en muchísimos otros juegos, se deberá llevar un recuento del número de vidas que le quedan al jugador. Lo que diferencia a unos juegos de otros es cómo gestionan el momento en el que el jugador se queda sin vidas. Muchos de ellos obligan al jugador a comenzar de nuevo, lo cual es muy típico en juegos con una historia muy larga, donde el progreso es continuado. Sin embargo en juegos con niveles, independientes unos de otros no tiene mucho sentido hacer eso, por lo que últimamente está muy de moda que el usuario deba esperar un determinado tiempo para que se recarguen sus vidas o también tienen la opción de pedírsela a sus amigos vía red social. Vistas estas opciones, queda claro que ninguna sirve para el propósito de la aplicación que aquí tratamos. Por lo que decidimos, que el jugador tenga que responder preguntas tipo test referentes al contenido de la asignatura. De forma que si la respuesta es correcta, se le dará al jugador una vida y si la respuesta es incorrecta se le propondrá otra pregunta diferente.

7. **Botones extra:**

   La aplicación debe de constar de una serie de botones que hagan más sencilla la interacción del usuario con la aplicación. De esta forma, el usuario podrá salir del juego cuando quiera, cargar de nuevo el nivel en plena partida, volver al menú de selección de niveles o cerrar la sesión.

3.5. Diagrama UML de las clases

En la **Figura 3.6** se representa mediante un diagrama UML la organización de las clases de las que consta la aplicación.

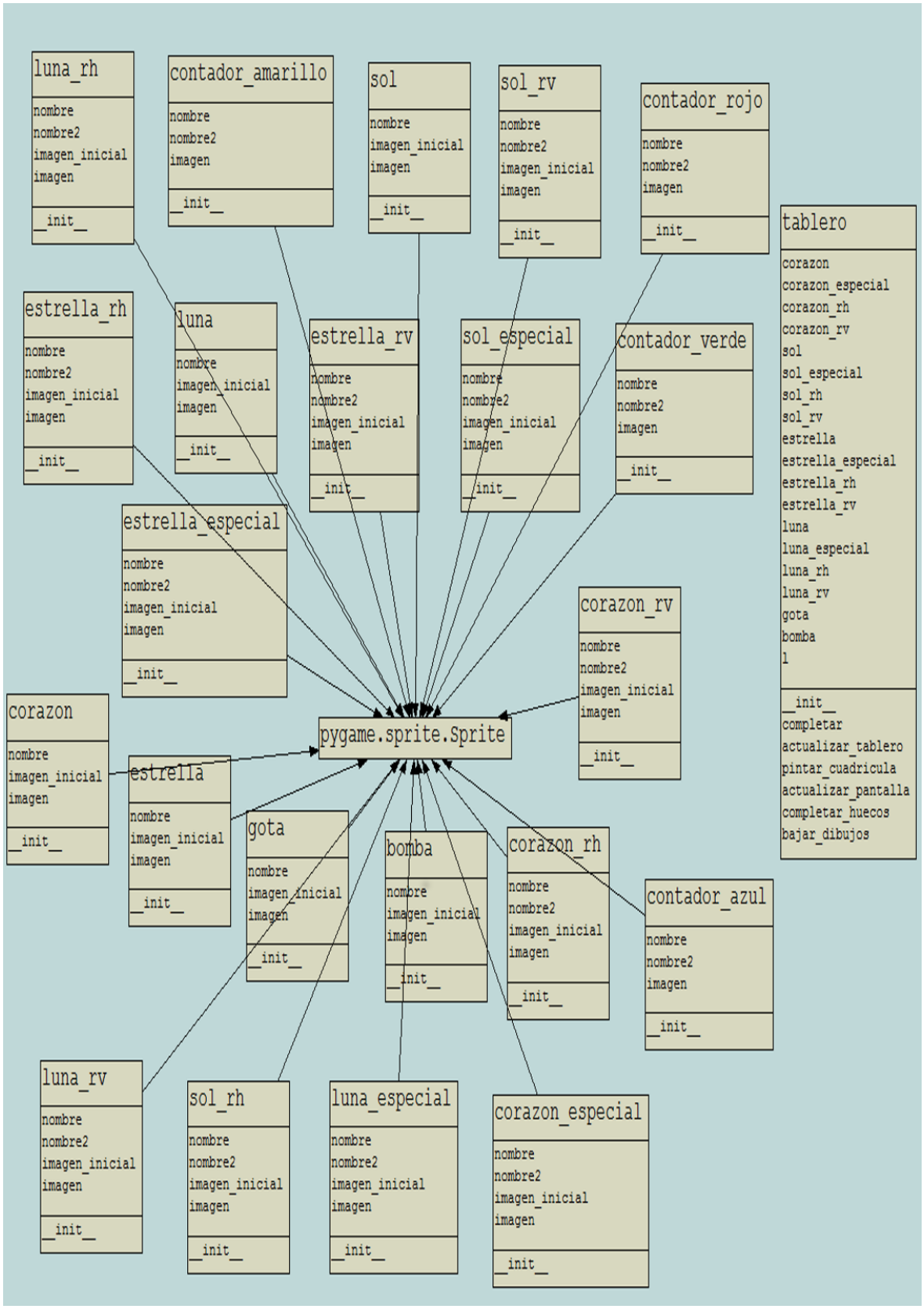

Figura 3.6: Diagrama uml de las clases.

Mediante el mencionado diagrama UML se pueden observar una serie de clases de tipo sprite. Cada una de estas clases representa un elemento de nuestro juego. Con esto me refiero a cada una de las fichas o figuras que pueden aparecer en nuestro tablero.

Todas ellas constan de una serie de atributos, usados para acceder a algunas características del sprite, tales como su nombre o su imagen. Ademas constan de un método "init()" que se ejecuta cuando instanciamos una de estas clases. Es decir, cada vez que queremos crear un objeto del tablero, basta con instanciar la clase correpondiente y el método "init()" asignará los parámetros correspondientes a los atributos del objeto.

Por otro lado, también se puede observar en la **Figura 3.6** otra clase llamada "tablero", la cual tiene como atributos todos los posibles objetos o fichas que pueden colocarse en el tablero del juego. Es decir, la función "init()" de la clase tablero recibe como parámetros los objetos anteriormente creados y que son instancias de las clases anteriormente nombradas. Además esta clase "tablero" tiene una serie de funciones propias que se usarán para realizar diferentes acciones sobre el tablero en general, basadas en la movilidad de las fichas principalmente.

3.6. Diagramas de flujo del funcionamiento de la aplicación

A continuación se va a exponer el funcionamiento de la aplicación, es decir, de los pasos que sigue el usuario una vez ejecutada la aplicación. De dichos pasos depende el flujo de los datos de la aplicación, lo cual representaremos mediante diagramas de flujo.

3.6.1. Inicio de la aplicación.

En primer lugar, obviamente el usuario debe ejecutar la aplicación, momento en el que se le presentará la pantalla en la que debe decidir si registrarse o si iniciar sesión. Elegirá registrarse si es la primera vez que juega e iniciar sesión si ya lo ha hecho antes.

En la **Figura 3.7** se muestra el diagrama de flujo que sigue la aplicación nada más iniciada.

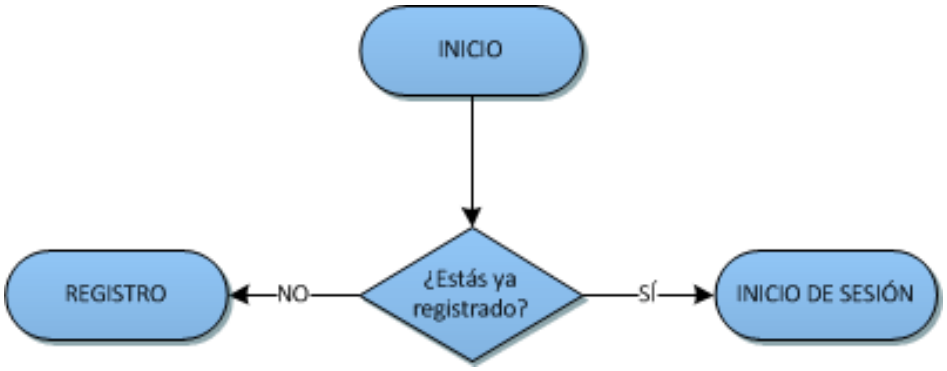

Figura 3.7: Diagrama de flujo del inicio de la aplicación.

3.6.2. Registro.

Si el usuario decide registrarse en la pantalla de inicio, entonces la aplicación fluirá por el siguiente camino.

- Se le pedirá al usuario que introduzca mediante el teclado el nombre de usuario que haya elegido y una contraseña. Ambos parámetros los introducirá en la correspondiente ventana para tal fin.

- A continuación, la aplicación consultará la base de datos, comparando el nombre del usuario introducido con los otros nombres que pudiera haber ya registrados en dicha base de datos.

- Fruto de esta consulta surgen dos caminos posibles por los que irá la aplicación.

- Si el nombre elegido por el usuario ya existe, es decir que ya fue elegido anteriormente por otro usuario, entonces se informará al usuario del error y se le reconducirá al inicio de la aplicación de nuevo, donde volverá a elegir entre registrarse o iniciar sesión.

- En caso contrario, si el nombre de usuario es correcto, se grabará en la base de datos junto con la contraseña y otros parámetros, como el número de vidas inicial y el nivel actual. Por último, una vez completado el registro correctamente se mostrará el menú con el primer nivel desbloqueado, puesto que es el único al que de momento puede jugar.

Cabe destacar también que en el proceso de registro la aplicación lo primero que comprueba es si ya se han creado con anterioridad las tablas para la base de datos en el dispositivo actual. Ya que en caso de que no existan, significa que el usuario es el primero en jugar en dicho dispositivo, de manera que no tiene que comprobar si hay coincidencia con el nombre de usuario y lo que sí debe de hacer antes de inscribirle es crear todas las tablas, de la base de datos, que en esta aplicación se utilizan.

En la **Figura 3.8** se muestra el diagrama de flujo que sigue la aplicación para que el usuario pueda completar el registro.

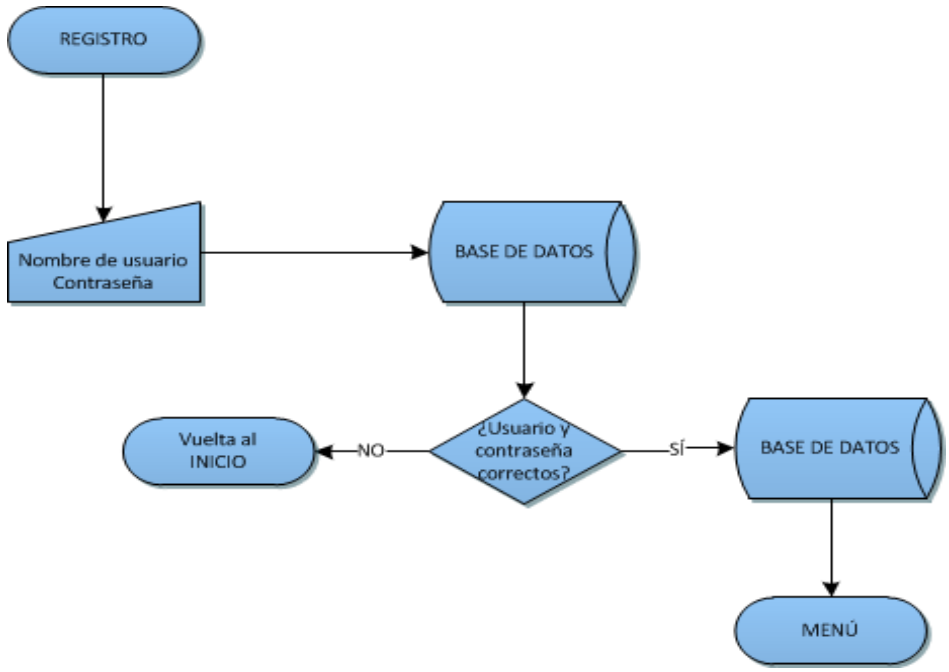

Figura 3.8: Diagrama de flujo del registro.

3.6.3. Inicio de sesión.

Por otro lado, si el usuario decide iniciar sesión, será porque ya haya jugado anteriormente y realizado el registro. El diagrama de flujo del inicio de sesión es muy parecido al del registro. La principal diferencia reside en las consultas a la base de datos.

- El usuario también tendrá que ingresar el nombre de usuario y la contraseña.
- A continuación, la aplicación consultará la base de datos y buscará en la tabla correspondiente un nombre de usuario y una contraseña que coincidan con las introducidas por el usuario.
- Si el nombre de usuario o la contraseña no son correctos, entonces se informará al usuario del error y se le rediccionará de nuevo al inicio de la aplicación, donde deberá intentar de nuevo el inicio de sesión.
- Si la búsqueda es satisfactoria, entonces se consultará de nuevo la base de datos. En esta nueva consulta se accederá al campo que contiene la información a cerca del último nivel superado por el jugador.
- Por último, se cargará el menú correspondiente, consecuentemente con el dato capturado de la base de datos a cerca del último nivel superado por el jugador. De esta manera, el jugador sólo podrá acceder a los niveles que ya haya superado y al inmediatamente superior al último superado.

En la **Figura 3.9** se muestra el diagrama de flujo que sigue la aplicación para que el usuario pueda completar el registro.

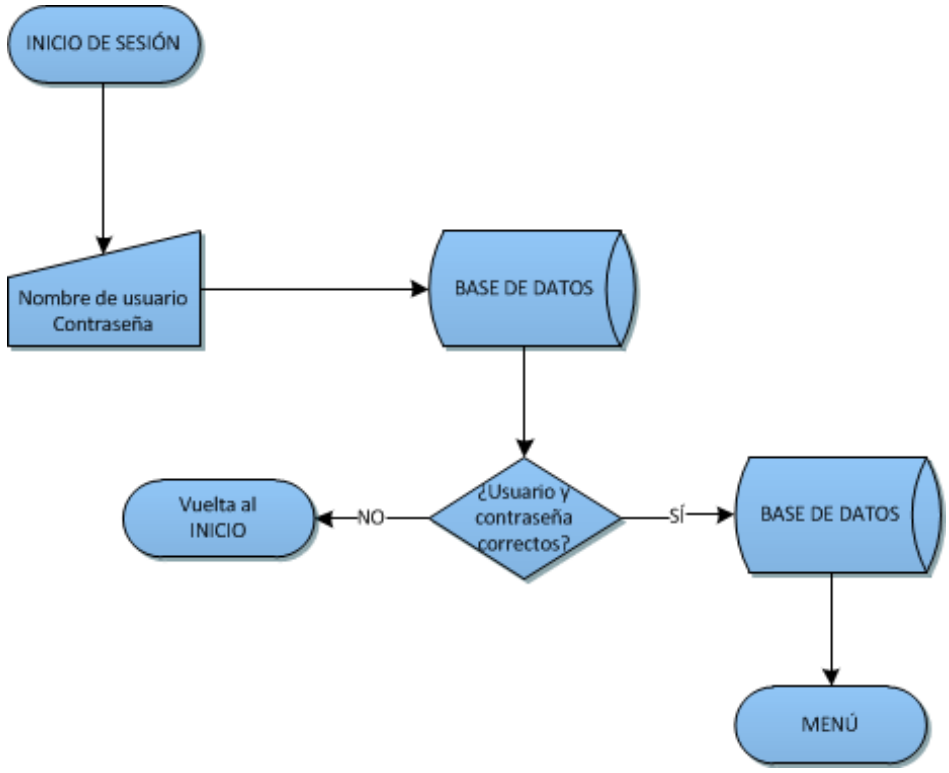

Figura 3.9: Diagrama de flujo del inicio de sesión.

3.6.4. Menú.

A continuación, se explica el diagrama de flujo correspondiente a la pantalla del menú, explicando las alternativas que tiene el usuario y como fluye el programa en cada elección.

- En primer lugar, el usuario podrá hacer básicamente tres cosas en la pantalla del menú, elegir el nivel al que desea jugar, ver las puntuaciones de cada nivel o cerrar la sesión.

- Si el usuario decide cerrar la sesión, se le redireccionará a la pantalla de inicio de la aplicación. Esta opción se presenta principalmente por si otro usuario decide jugar a continuación del anterior, de manera que no tenga que cerrar la aplicación y volver a ejecutarla.

- Por otro lado, si el usuario decide pulsar el botón correspondiente para ver la clasificación accederá a una nueva pantalla en la que visualizará una tabla. En dicha tabla estarán reflejadas las puntuaciones de cada nivel que los diferentes usuarios han superado. Es básicamente una visualización de la tabla llamada "historial" almacenada en la base de datos, de la cual hablaremos con mayor profundidad más adelante. En esta nueva pantalla de visualización de las puntuaciones, el jugador únicamente podrá pulsar el botón de regresar al menú, a parte obviamente del botón para cerrar la aplicación, el cual se encuentra presente en todas las pantallas de la aplicación.

- Si por contra el usuario decide jugar a uno de los niveles que tiene desbloqueados, accederá a otra pantalla previa al inicio del nivel. En esta pantalla se le presentará al jugador una breve descripción del nivel, presentándole los retos que debe conseguir para superar el nivel.

- En esta pantalla intermedia, el usuario podrá arrepentirse del nivel al que iba a acceder y regresar al inicio del menú, para poder elegir otro nivel.

- Además el usuario, obviamente podrá elegir jugar a dicho nivel, de manera que se dará el inicio de la partida.

En la **Figura 3.10** se muestra el diagrama de flujo que sigue la aplicación a través del menú.

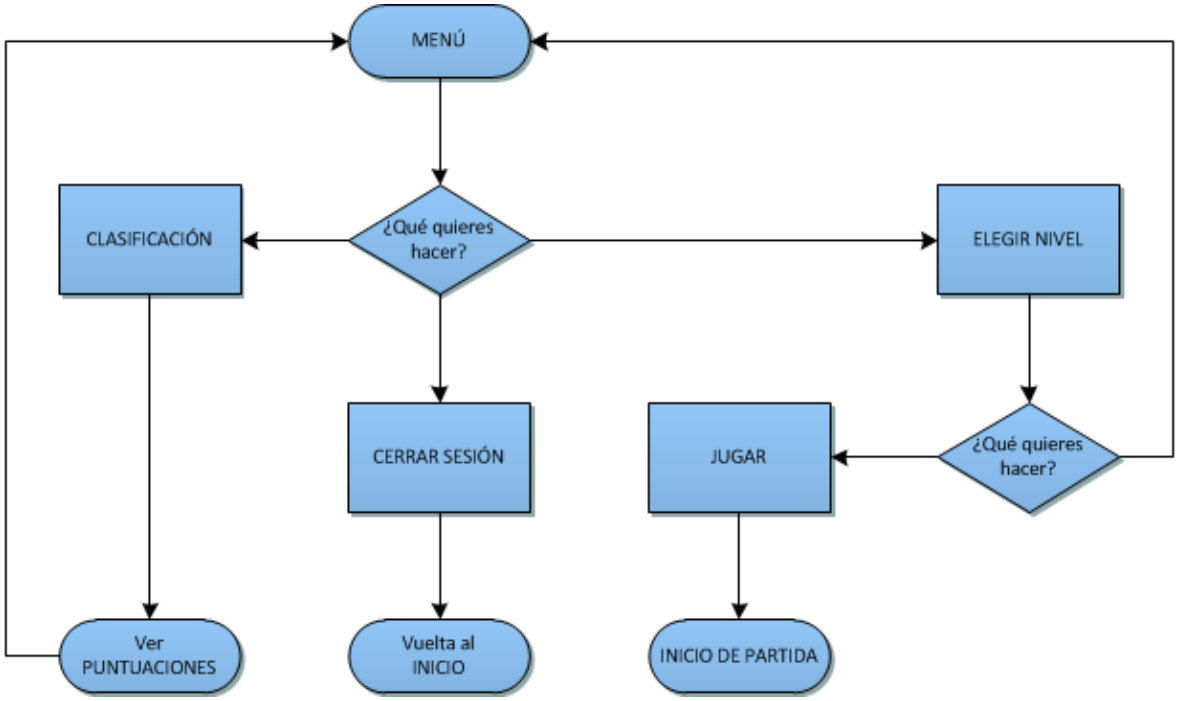

Figura 3.10: Diagrama de flujo del menú.

3.6.5. Inicio de partida.

En este caso se explicará el diagrama de flujo correspondiente a la pantalla en la que el jugador ha accedido al nivel y ha comenzado la partida.

- El jugador va a tener dos caminos posibles, o reiniciar la partida o disputar la partida.
- Si el jugador decide reiniciar la partida, automáticamente se cargará de nuevo el nivel, con una nueva distribución del tablero obviamente. Además cuando el jugador decide reiniciar la partida, habrá un cambio en la base de datos, ya que habrá que quitarle una vida al jugador, puesto que reiniciar la partida es lo mismo que abandonarla o darla por perdida.
- Por el contrario, si el jugador decide disputar la partida hasta el final, como en todo juego puede ocurrir dos cosas, que se produzca una derrota o que consiga la victoria, en función de que haya conseguido alcanzar los objetivos que se le presentaron previamente para el nivel correspondiente.

En la **Figura 3.11** se muestra el diagrama de flujo que sigue la aplicación una vez iniciada la partida.

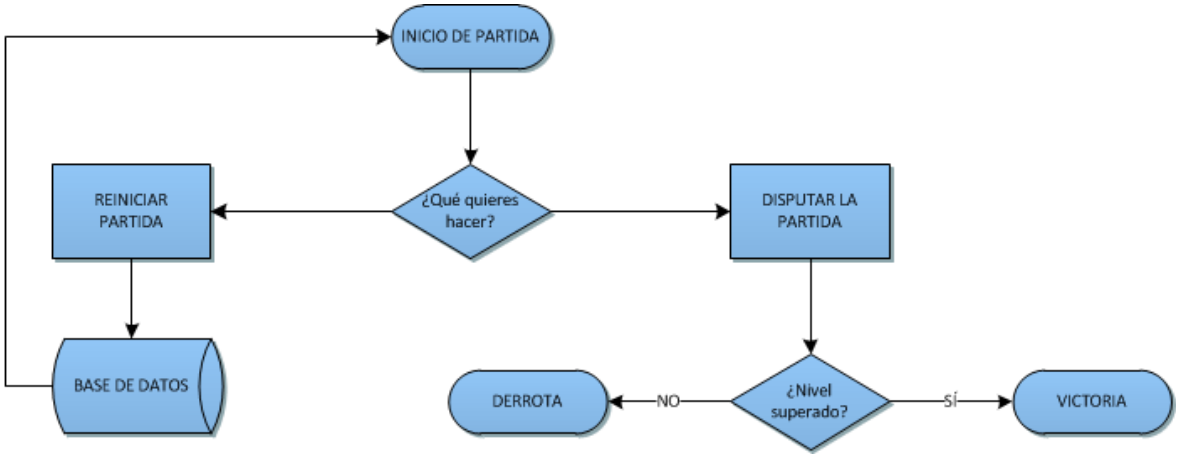

Figura 3.11: Diagrama de flujo del inicio de la partida.

3.6.6. Victoria.

El diagrama de flujo que sigue el programa con la victoria de un nivel aparece reflejado en la **Figura 3.12** y se explica a continuación.

- En primer lugar, aparece una pantalla en la que se informa al jugador de la victoria.
- Además se produce un acceso a la base de datos. En este acceso, se comprueba si el nivel que acaba de superar había sido jugado anteriormente. En caso que sea así, sólo se guardará la puntuación de la última victoria en caso de ser superior que la puntuación que ya figura en la base de datos y que corresponde a una partida pasada. Por el contrario, si el jugador es la primera vez que supera dicho nivel, se guardará directamente la puntuación recientemente obtenida en dicho nivel.
- Por otro lado, el jugador tendrá dos opciones, volver a jugar el nivel o volver al menú para poder jugar otro nivel.

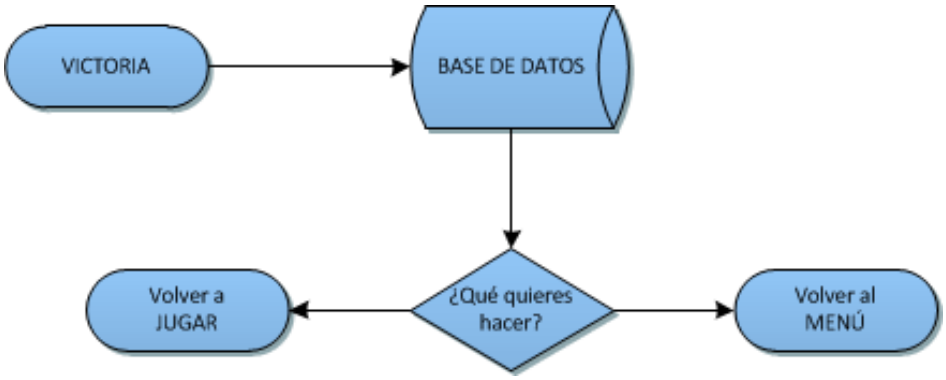

Figura 3.12: Diagrama de flujo de la victoria.

3.6.7. Derrota.

A continuación se explica cómo funciona el programa cuando el jugador sufre una derrota al jugar a un determinado nivel.

- En primer lugar, una vez que tiene lugar la derrota, se muestra una pantalla en la cual se informa de la derrota del nivel, a la vez que se le resta una vida al jugador en la base de datos. A partir de aquí, el jugador tiene dos opciones.
- La primera consiste en volver al menú si desea jugar a otro nivel.
- En segundo lugar, si el usuario decide jugar otra vez el programa consultará la base de datos. En esta consulta comprueba el número de vidas que tiene el jugador.
- Si el jugador se ha quedado sin vidas, autómaticamente el programa accede a la pantalla en la que se lanza una pregunta de test, de verdadero y falso, a la que el jugador debe responder.
- Por el contrario, en caso que el jugador aún disponga de vidas, se reiniciará el nivel actual.

En la **Figura 3.13** se muestra el diagrama de flujo que sigue la aplicación una vez que el jugador sufre una derrota en el nivel.

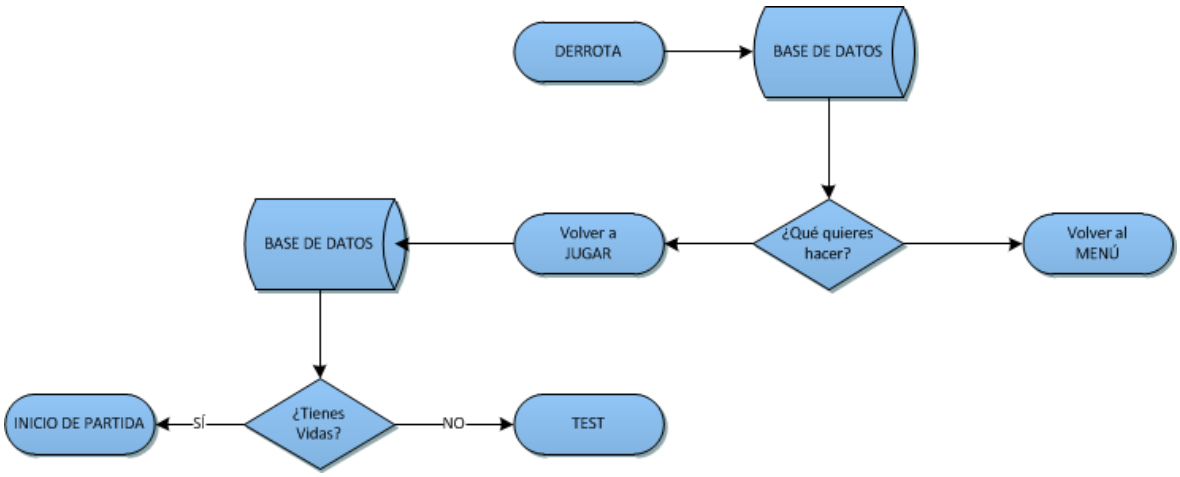

Figura 3.13: Diagrama de flujo de la derrota.

3.6.8. Test.

Por último, se explica a continuación el diagrama de flujo que sigue la aplicación cuando el jugador quiere seguir jugando y no tiene vidas, de manera que entra a la pantalla de test.

- En primer lugar, se consulta la base de datos para cargar una de las preguntas que en ella se encuentran grabadas y se le muestra al usuario, que tendrá que decidir si la sentencia es verdadera o falsa.

- Si la respuesta del usuario es incorrecta, se grabará su respuesta como errónea asociada a la pregunta en la base de datos y automáticamente se volverá al inicio del test, lanzándole otra pregunta al jugador.

- Si por el contrario la respuesta es correcta se accederá a la base de datos para realizar varias acciones. Por un lado se grabará el estado de la respuesta como acertado y además se sumará una vida al registro de vidas del jugador. Además se consultará el último nivel superado por el jugador y se le redireccionará al menú, cargando el que corresponda en cada caso.

  En la **Figura 3.14** se muestra el diagrama de flujo que sigue la aplicación una vez que se accede al test.

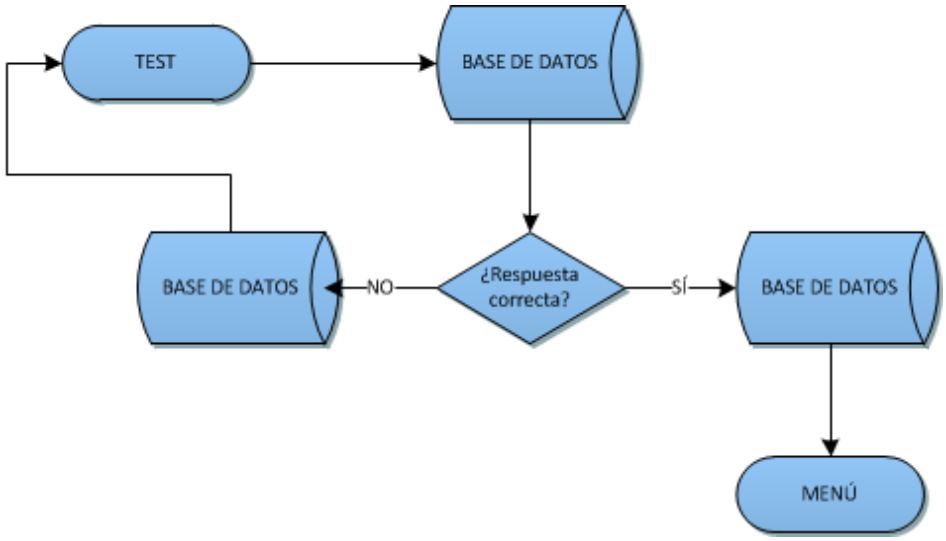

Figura 3.14: Diagrama de flujo del test.

3.7. Gestión de los datos

Un aspecto muy importante que debe tratar toda aplicación es cómo va a ordenar sus datos. Las aplicaciones usan datos, que suponen información de muchos tipos y que utilizan para cosas muy diversas. Además estos datos pueden ser fijos o estar cambiando continuamente durante la ejecución de la aplicación. Por todo esto, las aplicaciones hacen uso de las bases de datos, en las que pueden ordenar los datos de manera segura, acceder a ellos cuando quieran y modificarlos cuando sea preciso.

En la aplicación desarrollada para este trabajo de fin de grado se ha usado una base de datos que consta de cuatro tablas.

Cada una de las tablas almacena datos que tienen una relación y cada una de ellas será útil para la aplicación en algún momento concreto de la ejecución de la misma.

Las tablas finalmente están formadas por un conjunto de registros o filas, y éstas por un conjunto de campos o columnas.

A continuación explicaré las tablas usadas en esta aplicación, detallando cada uno de los campos de los que están formadas.

1ª TABLA: *registro*

En esta tabla, representada en la **Figura 3.15**, se almacenan los datos correspondientes al registro y al inicio de sesión del usuario.

usuario	contraseña	ultimo_nivel	vidas
kike	kike	2	1
guillermo	guille	0	3
eric	cabeza	1	2
pablillo	lagar	0	1

Figura 3.15: Tabla del registro.

Consta de los siguientes campos:

- **usuario:** almacena el nombre de usuario elegido por el jugador en el registro. No debe coincidir con ningún otro nombre de usuario que se encuentre ya inscrito en esta misma tabla. Éste campo también será usado por la aplicación cuando el usuario quiera iniciar sesión, ya que debe coincidir con el introducido por el usuario en ese momento.
- **contraseña:** almacena la contraseña elegida por el usuario en el momento del registro. En éste campo, a diferencia del nombre de usuario, no se comprueba si ya existe otra contraseña igual, ya que en este caso no es relevante. El campo de la contraseña también se utilizará en el procedimiento de inicio de la sesión, ya que debe coincidir con la contraseña introducida por el usuario en ese momento.
- **ultimo_nivel:** en este campo se almacena el número correspondiente al último nivel que ha superado el usuario. Esta información será requerida por la aplicación cuando el usuario acceda al menú del juego, ya que en ese momento sólo le aparecerán disponibles los niveles que ya haya superado y el inmediatamente superior, es decir tantos niveles como indique este campo y además el siguiente nivel. Éste campo será modificado cada vez que el usuario complete un nuevo nivel.
- **vidas:** en este último campo de la tabla se almacena el número correspondiente a las vidas que le quedan al jugador para poder seguir jugando antes de tener que enfrentarse a una pregunta de test. Se accederá por lo tanto a este campo cada vez que el usuario pretenda jugar a un nivel, en caso de quedar vidas podrá completar la acción. Por el contrario si no le quedan vidas deberá responder a una pregunta de test. En consecuencia, si acierta la pregunta de test, el campo de las vidas se volverá a modificar, ya que se premierá el acierto dando una vida extra al usuario. Este campo de las vidas, también sufrirá una modificación, en este caso negativa, cada vez que en mitad de una partida reinicie el nivel o salga de la aplicación.

2ª TABLA: *historial*

En esta tabla, la cual ya fué nombrada con anterioridad en el apartado 3.6.4 y que aparece representada en la **Figura 3.16**, se guardará un registro de los niveles superados por los usuarios.

usuario	nivel	puntos
kike	1	150000
kike	2	210000
eric	1	158620

Figura 3.16: Tabla del historial de puntuaciones.

Consta de los siguientes campos:

- **usuario:** igual que en la tabla anterior este campo almacena el nombre de usuario del jugador, pero a diferencia de la anterior tabla no es el campo consultado cuando se quiere comprobar la identidad del usuario. Este campo es usado únicamente para guardar una relación entre el usuario y los niveles superados.
- **nivel:** como se ha mencionado en la explicación del campo anterior, en este campo se guarda un número correspondiente a un nivel. Para que quede inscrito el registro en esta tabla, dicho nivel debe haber sido superado por el jugador.
- **puntos:** en este campo se almacena un número que corresponde a la puntuación con la que el usuario ha superado el correspondiente nivel.

Por lo tanto, en esta tabla, cada registro o entrada relaciona a un nombre de usuario con un nivel superado y la puntuación correspondiente.

Cada una de las entradas puede suponer una nueva entrada o una actualización de una ya existente. Las nuevas entradas tendrán lugar cuando el jugador sea la primera vez que supera el nivel. Por otro lado, si el jugador ya hubiera superado el nivel con anterioridad, entonces sólo se registrará la entrada actualizando la puntuación en caso de que la puntuación actual sea superior a la ya registrada anteriormente.

Finalmente queda claro con lo anteriormente explicado, que cuando el jugador no supera el nivel no queda constancia de ello, ni tampoco cuando lo supera con una puntuación inferior a la vez anterior.

3ª TABLA: *preguntas*

En esta tabla, que aparece representada en la **Figura 3.17**, se guarda un registro de las preguntas de test que posteriomente se lanzarán al jugador para que pueda recuperar vidas.

id	sentencia	respuesta
1	"enunciado"	"verdadero"
2	"enunciado"	"falso"
3	"enunciado"	"falso"

Figura 3.17: Tabla con las preguntas de test.

Consta de los siguientes campos:

- **id:** en este campo se almacena un número entero que corresponde al número de la pregunta que se le asigna a una determinada sentencia.
- **sentencia:** este campo almacena el texto correspondiente a la pregunta que se le formulará al jugador cuando corresponda. En la imagen de la tabla se ha sustituido el texto de la sentencia por la palabra «enunciado». Un ejemplo de sentencia, a la que el jugador debe responder con verdadero o falso, sería el siguiente:

  "Un programa en C es un conjunto de definiciones de tres tipos: variables,funciones y tipos de datos."
- **respuesta:** En este campo se almacena la palabra «verdadero» o «falso». Es decir, este campo almacena la respuesta a la pregunta que se le ha formulado al jugador y que aparece almacenada en el campo llamado "sentencia" de la anterior tabla. Para el ejemplo propuesto anteriormente la respuesta correcta sería: "verdadero".

Esta tabla será consultada por la aplicación cuando el usuario se quede sin vidas. En este momento, cargará de forma aleatoria una de las sentencias almacenadas en dicha tabla.

4ª TABLA: *cuestionario*

En esta tabla, que aparece representada en la **Figura 3.18**, se almacena un registro de las respuesta efectuadas por el usuario.

usuario	id	estado
kike	1	"BIEN"
pablo	4	"MAL"
eric	3	"BIEN"
kike	3	"MAL"

Figura 3.18: Tabla con las respuestas de cada alumno.

Consta de los siguientes campos:

- **usuario:** como en ocasiones anteriores guarda el nombre de usuario, que vuelve a ser fundamental para relacionarle con sus respuestas. También es el campo utilizado cuando se quiere acceder a esta tabla, ya que la consulta más lógica es ver el registro de respuestas de un determinado jugador.
- **id:** en este campo se almacena un número entero que hace referencia al número de sentencia almacenada en la tabla anterior.
- **estado:** en este campo se guarda si la respuesta del usuario a la pregunta que se corresponde con el «id» del campo anterior es correcta o incorrecta. De esta forma se almacenará la palabra «BIEN» cuando el usuario haya contestado correctamente a la pregunta y la palabra «MAL» cuando la respuesta haya sido incorrecta.

3.8. Resumen del capítulo

En el presente capítulo se han explicado los pasos que se han seguido en el análisis, diseño y desarrollo de la aplicación que conforma el presente trabajo de fin de grado.

En primer lugar se han explicado un conjunto de programas que han intervenido en la elaboración de distintos aspectos dentro del trabajo de fin de grado. Aspectos como el manejo de bases de datos, redacción de la presente memoria, elaboración de diagramas de flujo o diagramas UML y elaboración del diagrama de gantt para la planificación del proyecto.

En los siguientes apartados se ha detallado en qué consiste el juego, explicando la dinámica del mismo y la creación de sus niveles.

También se han detallado los diferentes requisitos que debe cumplir la aplicación para satisfacer los objetivos de la misma.

Por otro lado, haciendo uso de un diagrama UML se ha explicado la relación existente entre las clases presentes en la programación del juego. Además, mediante diagramas de flujo se ha explicado el funcionamiento de la aplicación, detallando las diferentes posibilidades de interacción entre el usuario y dicha aplicación.

Por último, se ha detallado la gestión de los datos que maneja la aplicación. En este caso se ha hecho uso de cuatro tablas creadas mediante SQLite, un sistema de gestión de bases de datos relacional.

Capítulo 4

Validación de requisitos

4.1. Validación

En este capítulo se van a revisar los requisitos establecidos en el capítulo anterior, los cuales deben ser satisfechos por la aplicación. Por ello, se argumentará, ayudándose de gráficos cuando proceda, la forma en la que dichos requisitos han sido tenidos en cuenta y llevados a cabo.

Los requisitos son los siguientes:

1. **Registro de usuario:**
   En la **Figura 4.1** se muestra la pantalla inicial del videojuego, en la cual el jugador debe seleccionar que quiere hacer, si registrarse, por no tener ya un perfil creado, o si quiere iniciar sesión, en caso de ya haberse registrado con anterioridad. En este caso, obviamente hará click sobre el botón «Registrarse».

Figura 4.1: Pantalla de inicio.

En la **Figura 4.2** se muestra la pantalla en la cual se le permite al jugador introducir el nombre de usuario deseado, el cual en este caso, por tratarse del registro de usuario, debe ser nuevo, es decir, no haber sido elegido anteriormene por otro usuario.

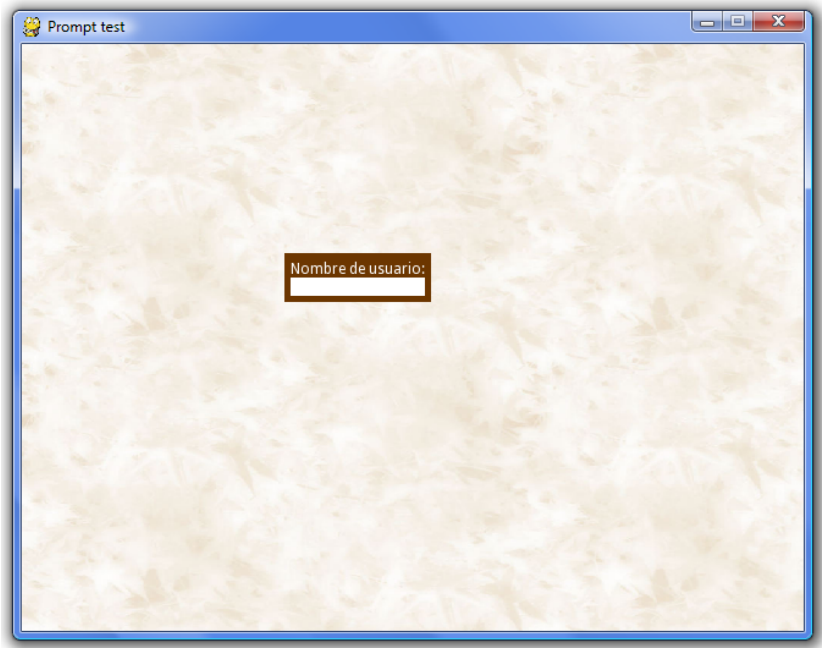

Figura 4.2: Introducir el nombre de usuario.

En la **Figura 4.3** se muestra la pantalla en la cual se le permite al jugador introducir la contraseña elegida para el nuevo perfil de usuario que está creando, y con la cual se le permitirá el acceso en ocasiones posteriores mediante la opción de inicio de sesión.

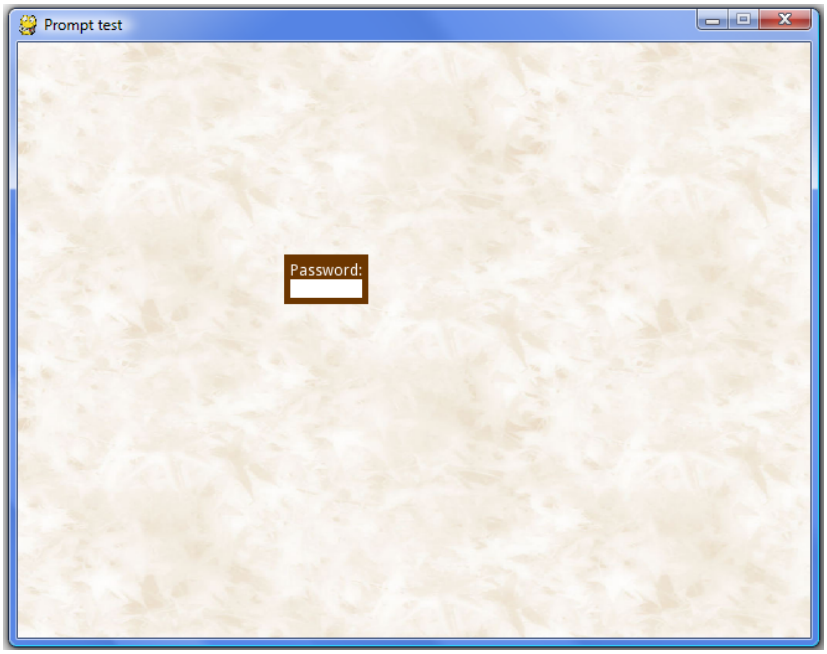

Figura 4.3: Introducir la contraseña.

En la **Figura 4.4** se muestra la pantalla que informa al jugador que su registro se ha realizado con éxito, es decir, que el nombre de usuario elegido no se ha sido asignado previamente a ningún otro jugador.

Figura 4.4: Registro realizado con éxito.

En la **Figura 4.5** se muestra la pantalla en la cual se informa al jugador que no se ha completado el registro correctamente, ya que el nombre de usuario elegido ya ha sido asignado previamente a otro jugador.

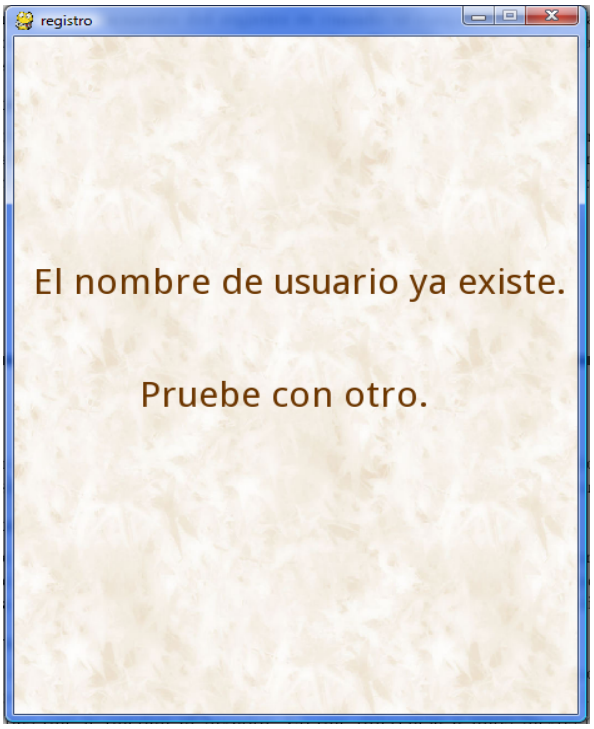

Figura 4.5: Registro incorrecto.

2. Inicio de sesión

Cuando el usuario intenta acceder por medio de la opción de inicio de sesión, la aplicación buscará en primer lugar el nombre de usuario y una vez que lo encuentra comprobará que la contraseña es correcta. En caso de que alguno de los dos parámetros no coincida, la aplicación informará al usuario del error y le dirá en cual de los dos parámetros se ha producido dicho error.

En la **Figura 4.1** se muestra la pantalla inicial del videojuego, en la cual el jugador debe seleccionar qué quiere hacer, si registrarse, por no tener ya un perfil creado, o si quiere iniciar sesión, en caso de ya haberse registrado con anterioridad. En este caso, para validar el presente requisito, el jugador pulsará sobre el botón «Iniciar sesión».

En la **Figura 4.2**, como ya se explicó en el requisito anterior, se le permite al jugador introducir su nombre de usuario, que en este caso debe coincidir obviamente con el elegido en el momento del registro, para así de esta forma poder acceder a su progreso en el videojuego.

En la **Figura 4.3**, como también se explicó en el requisito anterior, se le permite al jugador introducir la contraseña elegida en el registro y que siendo correcta le permitirá acceder al videojuego.

En la **Figura 4.6** se muestra la pantalla en la cual se informa al jugador que ha iniciado su sesión correctamente, es decir que ha introducido correctamente tanto el nombre de usuario como la contraseña.

Figura 4.6: Inicio de sesión correcto.

En la **Figura 4.7** se muestra la pantalla en la cual se informa al jugador que el nombre de usuario introducido es incorrecto, por lo que debe intentar de nuevo el inicio de sesión.

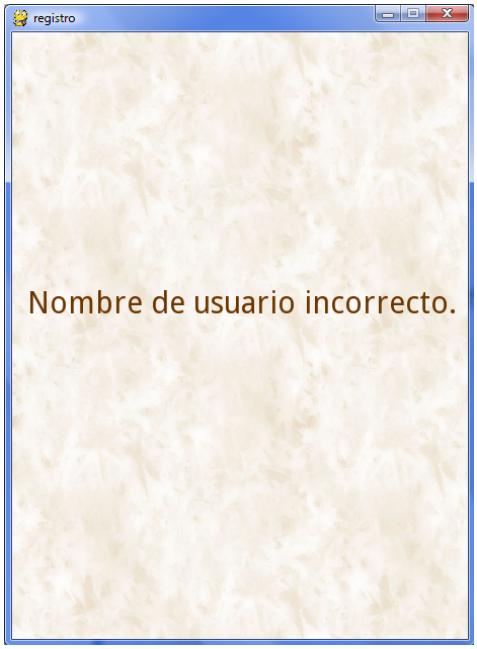

Figura 4.7: Nombre de usuario incorrecto en el inicio de sesión.

En la **Figura 4.8** se muestra la pantalla en la cual se informa al jugador que la contraseña introducida es incorrecta, por lo que debe intentar de nuevo el inicio de sesión.

Figura 4.8: Contraseña incorrecta en el inicio de sesión.

3. **Menú de los niveles:**

   En la **Figura 4.9** se muestra el menú que corresponde al progreso particular del jugador, en este caso se trata de un jugador principiante que aún no ha superado el nivel 1, por lo que es el único nivel que tiene disponible para jugar.

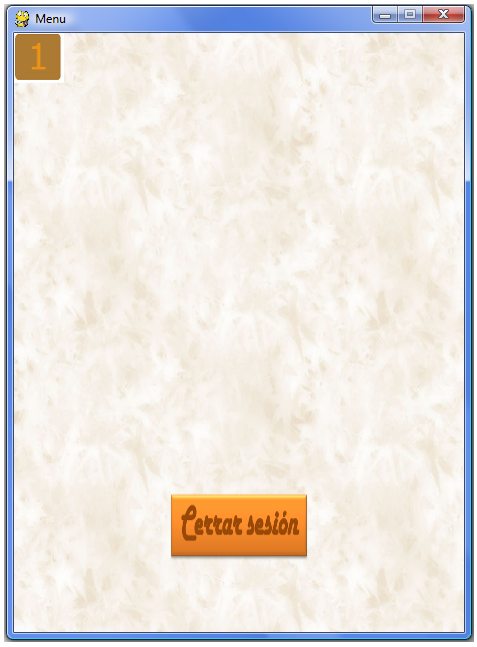

Figura 4.9: Menú con 1 nivel desbloqueado.

En la **Figura 4.10** se muestra el menú que corresponde al progreso particular del jugador, en este caso se trata de un jugador que ya ha superado el primer nivel del juego, por lo que en su menú aparecen desbloqueados los dos primeros niveles del juego. De esta misma forma seguirán apareciendo los botones de entrada a los niveles correspondientes según el jugador vaya superando niveles.

Figura 4.10: Menú con 2 niveles desbloqueados.

4. **Ver clasificación:**

En la **Figura 4.11** se muestra la pantalla de clasificación de los usuarios, con la puntuación obtenida en cada uno de los niveles que han superado.

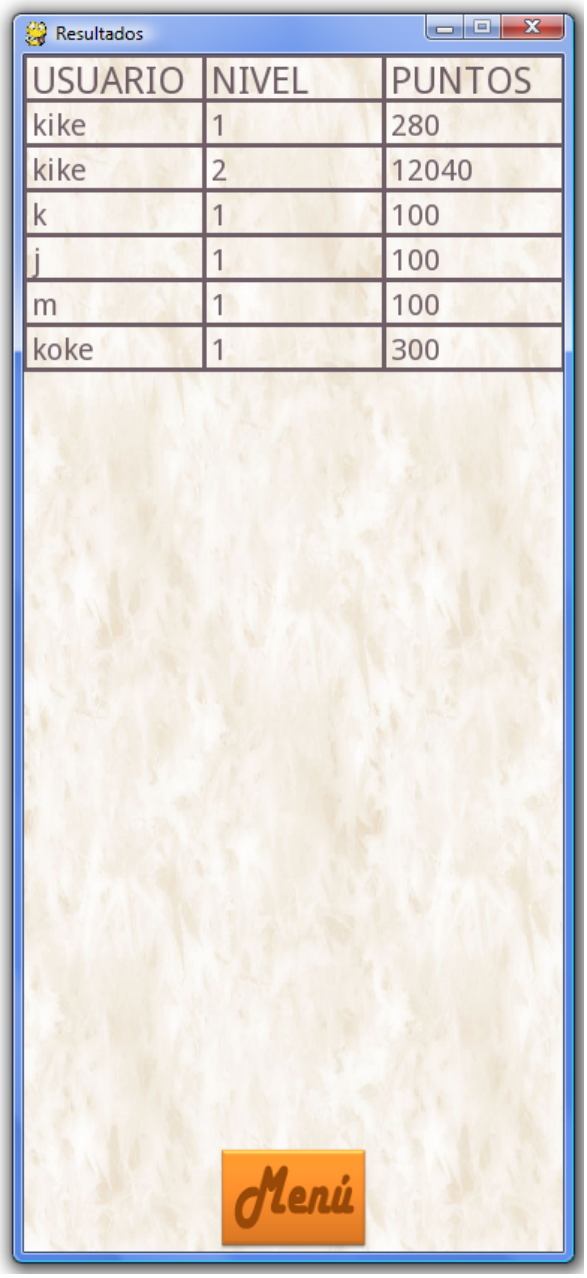

Figura 4.11: Clasificación.

5. Descripción del nivel

En la **Figura** 4.12 se muestra la pantalla previa a disputar un nivel. En esta pantalla se muestran los retos propios a dicho nivel, los cuales deben ser superados por el usuario. En la mencionada figura se muestra en concreto la pantalla previa al nivel 2 y sus correspondientes retos.

Figura 4.12: Retos del nivel 2.

6. **Recuperar vidas respondiendo preguntas:**
   En la **Figura 4.13** se muestra el mensaje que informa al jugador que se ha quedado sin vidas y no puede seguir jugando de momento, de manera que tendrá que responder y acertar preguntas de verdadero o falso para recuperar vidas.

Figura 4.13: Jugador sin vidas.

En la **Figura 4.14** se muestra la pantalla en la que se le presenta al jugador una pregunta de test cuando ha perdido sus vidas. Debe responder verdadero o falso, en función de si cree que la sentencia propuesta es cierta o no lo es.

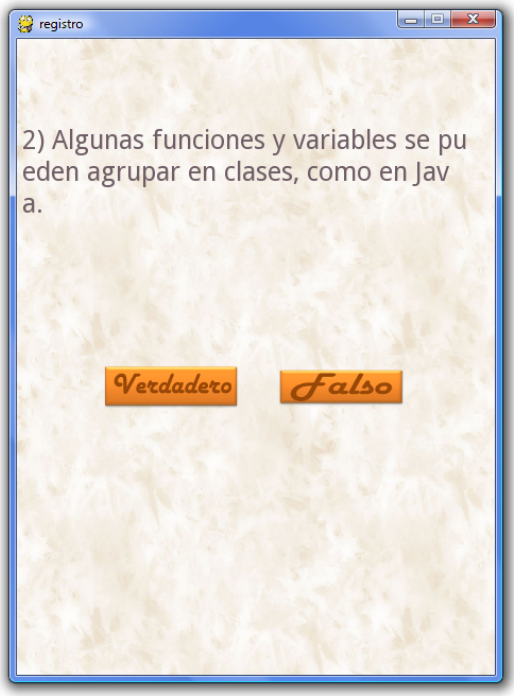

Figura 4.14: Pregunta de test.

En la **Figura 4.15** se muestra el mensaje que informa al jugador que ha respondido correctamente a la pregunta de test planteada y por lo tanto ha recibido una vida, de manera que puede continuar jugando.

Figura 4.15: El jugador consigue 1 vida.

7. Botones extra:

Como se puede observar en anteriores páginas en la **Figura 4.9** o en la **Figura 4.10** existe el clásico botón de la cruz en rojo en la parte superior

derecha de la pantalla, que sirve para cerrar la aplicación. Además se puede observar también en estas mismas imágenes el botón que permite al usuario cerrar la sesión, para que otro jugador pueda abrir la suya sin necesidad de cerrar por completo la aplicación.

Por otro lado, en la **Figura 4.16** se observa en la parte inferior de la pantalla un botón de color amarillo con una flecha circular. Éste botón es el de reiniciado del nivel, que permite al jugador en cualquier momento de la partida volver a cargar el nivel actual. El uso de este botón será típico cuando el jugador considere de antemano que no va a superar el nivel y prefiere volverlo a intentar desde el principio que seguir hasta el final. Obviamente el uso de este botón conlleva la pérdida de una vida para el jugador.

Figura 4.16: Reiniciar el nivel.

En la **Figura 4.17**, utilizada anteriormente en la validación de otro requisito, se puede observar un botón en el que se puede leer la palabra «Menú». Este botón sirve para regresar al menú de selección de nivel y aparece en varios momentos, siempre para facilitar al usuario su interacción con la aplicación.

Figura 4.17: Retos del nivel 2.

4.2. Conclusión

Una vez finalizado el presente capítulo ha quedado demostrado que todos los requisitos previamente establecidos en el capítulo **3** han sido cumplidos por la aplicación de manera satisfactoria. Dicha demostración se ha llevado a cabo a lo largo de este capítulo 4, haciendo uso de capturas de pantalla de la propia aplicación.

Capítulo 5

Conclusiones y trabajo futuro

En este capítulo se explicarán cuáles han sido las conclusiones extraídas una vez desarrollada la aplicación y los posibles desarrollos futuros para mejorar la misma.

5.1. Conclusión final

Quizá la conclusión final sobre el cumplimiento o no de ciertos objetivos habría que sacarla cuando la aplicación se pusiera en uso, ya que este sería el momento de ver si verdaderamente la aplicación puede formar parte de la docencia de una asignatura.

Sin embargo sí se pueden sacar otras conclusiones sobre aspectos técnicos de la propia aplicación. Para ello, debemos fijarnos en los objetivos establecidos al principio de la memoria, en el apartado **1.3** y hacer referencia a los mismos para sacar conclusiones:

- En cuanto al primer objetivo, se ha realizado un estudio, desarrollado en el estado del arte, de las diferentes herramientas que podrían haberse usado en el desarrollo del presente trabajo de fin de grado. Finalmete se optó por elegir Pygame como motor de desarrollo para el presente videojuego.
- En cuanto a los lenguajes de programación se eligió Python, el cuál a parte de no haberse impartido a lo largo de la carrera se encuentra en auge.
- En el presente trabajo de fin de grado se ha realizado el desarrollo de un prototipo de juego del tipo conecta-tres en dos dimensiones, llamado "Star-Crush", que permite al jugador enfrentarse a varios niveles con diferentes dificultades y retos.
- Eljuego pretende utilizarse como herramienta docente. Para ello se ha desarrollado un sistema de recuperación de vidas, que consiste en una serie de preguntas, con contenido teórico de la asignatura, a las que debe contestar el jugador cuando se quede sin vidas.
- La aplicación maneja una serie de datos, como pueden ser el nombre de usuario, la contraseña, las vidas del jugador o sus puntuaciones. Para manejar todos estos datos, se ha hecho uso de un sistema de gestión de bases de datos relacional, en concreto SQLite. Como se explicó anteriormente en el

apartado 3.7, se han creado cuatro tablas, cada una con una determinada relación, para el correcto manejo de dichos datos.

5.2. Trabajo futuro

A lo largo de esta memoria, se ha mencionado en algún momento que esta aplicación supone una primera implementación de una idea más grande. En el presente trabajo de fin de grado se ha mostrado como realizar una aplicación de escritorio. De esta forma los datos almacenados por la aplicación se guardarán de manera local en el ordenador donde se ejecute dicha aplicación. Por lo tanto, en este punto surge una de las implementaciones futuras más interesantes. Se trata de implementar la aplicación de manera que conecte con un servidor web, así de esta forma, todos los datos referentes al videojuego serán visibles por cada uno de los jugadores, sin que éstos tengan que jugar desde el mismo ordenador. Además, esta sería la forma más correcta para que la profesora pudiera ver la evolución de cada uno de sus alumnos en el videojuego y las respuestas al test efectuadas por cada uno de ellos. Además esta mejora, facilitaría el proceso de agregar preguntas de test, para ir renovando la aplicación a medida que avanza el contenido de la asignatura.

Por otro lado, otra vía que se debería seguir para mejorar la presente aplicación, es la de continuar implementando niveles. Cada nivel es independiente del anterior, por lo que sólo hace falta un programador con imaginación y de esta forma el presente videojuego no tiene porqué tener fin.

Estos dos posibles desarrollos presentados, bien podrían formar parte de un trabajo fin de grado futuro, que dejo en manos de algún alumno que como yo quiera mejorar sus conocimientos de programación. Obviamente para la segunda parte, seguir creando niveles, el programador primero tendrá que dedicar un tiempo a revisar y entender el código fuente que yo he creado, pero puede tomarse ese tiempo más que como una carga como un medio de aprender el lenguaje Python y Pygame a la vez. Además siempre viene bien complementar la información que se extrae de los manuales con casos prácticos y ejemplos donde se vea como aplicar las enseñanzas de los manuales, por lo que repito que la comprensión del código fuente de este trabajo de fin de grado se debe ver como un amplio ejemplo con el que mejorar los conocimientos en programación de videojuegos con las herramientas de Python y Pygame.

5.3. Conclusión personal

Me acerco al final de la redacción de la memoria y es el momento de sacar conclusiones, de echar la vista atrás y observar el trabajo realizado a lo largo de estos meses. El trabajo ha sido duro pero gratificante. El desconocimiento inicial, me hacía sentir desbordado, las dudas posteriores provocaban nerviosismo, pero cada reto conseguido, por pequeño que fuera, daba esperanzas e ilusión. Así es como poco a poco fuí ganando en confianza y ésta por fin trajo la motivación y la seguridad de que el esfuerzo daría sus frutos. Ahora, en el final, es el momento de afirmar y comprobar que esto es así, que el trabajo ha dado sus frutos y que me

encuentro cerrando la memoria de un trabajo de fin de grado que personalmente me ha dado mucho.

Muchas fueron las ideas previas y muchos los requisitos que cumplir. Todos ellos con la finalidad de realizar una aplicación educativa basada en la motivación que ocasionan los videojuegos en la sociedad y sobre todo en los jóvenes. Esta aplicación debía suponer el punto de partida para poder incluirla posteriormente en la dinámica de una asignatura de programación en C, es decir ser usada como herramienta docente.

En este trabajo de fin de grado se han presentado los pasos necesarios para realizar tal herramienta. De esta forma, la presente memoria puede servir de guía para otros, tanto si quieren crear su videojuego propio, como si prefieren modificar y mejorar la presente aplicación.

Cuando algo se termina se sacan conclusiones y éste es el momento. Para ello se revisan los objetivos a priori, se revisa lo que se quería conseguir antes de empezar y se compara con lo que finalmente se ha conseguido. De esta forma, a lo largo de la memoria se han ido presentando objetivos, requisitos y metas que conseguir, al igual que se ha ido demostrando como todos ellos se han cumplido. Principalmente se ha demostrado el cumplimiento de los requisitos técnicos que necesita cumplir la aplicación para así de esta forma satisfacer las necesidades que debe cubrir la misma. Lo que no se ha hecho y es más difícil de demostrar es un requisito personal que fue nombrado en la sección **1.3** y fue mencionado además como un objetivo muy importante para mí. Por si ya se ha olvidado, recuerdo que este objetivo era el de aprender y mejorar mis conocimientos de programación. Y llegados a este punto, no puedo decir otra cosa que no sea que este objetivo también se ha conseguido con éxito, pues todo el proceso de desarrollo de la aplicación, ha sido un camino enriquecedor, lleno de pequeños éxitos personales y de nuevos conocimientos adquiridos que me hacen sentir orgulloso de la etapa que comencé al empezar esta carrera y que me hacen sentir orgulloso e ilusionado por el futuro que se abre frente a mí, ahora que estoy llegando al final.

Apéndice A

Planificación y presupuesto

En esta parte de la memoria se procederá a explicar las diferentes etapas de las que se compone el desarrollo total del trabajo de fin de grado. Estas etapas son básicamente una serie de pasos que han sido seguidos y que garantizan el correcto cumplimiento de los objetivos y necesidades que la aplicación debe cubrir.

Además se detallará el presupuesto final del proyecto, fruto de las tareas llevadas a cabo así como del material empleado en el desarrollo del presente trabajo.

A.1. Planificación

La planificación se compone de una serie de tareas, que podemos analizar por separado. Estas tareas pueden ser independientes de las otras o por el contrario pueden depender de la realización previa de alguna otra.

En la tabla **A.1**, se muestran las principales tareas de las que ha constado el desarrollo del presente trabajo de fin de grado, especificando una fecha de inicio y de fin de la tarea aproximadas:

	Tarea	**Fecha de inicio**	**Fecha de fin**
1	Estudio de Python	1/2/2014	1/3/2014
2	Estudio de Pygame	1/2/2014	1/3/2014
3	Lógica del juego	15/2/2014	15/5/2014
4	Interfaz gráfica del juego	15/2/2014	15/7/2014
5	Estudio de MySQL	1/6/2014	5/6/2014
6	Implementación de base de datos en MySQL	6/6/2014	10/6/2014
7	Estudio de SQLite	15/7/2014	16/7/2014
8	Implementación de base de datos en SQLite	17/7/2014	18/7/2014
9	Aprendizaje de LaTeX	28/6/2014	30/6/2014
10	Redacción de la memoria	1/07/2014	30/07/2014
11	Presentación	8/09/2014	19/09/2014

Tabla A.1: Planificación dividida en tareas.

A continuación se explica brevemente en qué consiste cada una de las tareas reflejadas en la tabla **A.1**:

- **Tarea 1:** en un primer lugar, el primer paso es aprender el lenguaje de programación que nos va a permitir implementar la aplicación, en este caso se trata del lenguaje de programación Python.
- **Tarea 2:** en segundo lugar, se realizó un estudio de la biblioteca Pygame, empleada como motor para desarrollar la interfaz gráfica del videojuego.
- **Tarea 3:** posteriormente el siguiente paso, una vez aprendidas las herramientas que me iban a permitir desarrollar el videojuego, llegaba el momento de ponerse a programar la lógica del videojuego, es decir el código fuente que controla absolutamente todo lo que ocurre desde que se ejecuta hasta que se cierra la aplicación.
- **Tarea 4:** esta tarea está bastante relacionada con la anterior. Ya que la tarea 3 hace referencia a la inteligencia del videojuego, es decir que hace el programa cuando el usuario realiza alguna accion y por otro lado, esta tarea número 4 hace referencia a cómo la aplicación, utilizando Pygame, hace visible todo lo que gestiona la tarea 3. Es decir la tarea 3 gestiona internamente como fluye la aplicación y la tarea 4 muestra gráficamente lo anterior. Ambas tareas se programan conjuntamente en el código fuente de la aplicación.
- **Tarea 5:** una vez que la implementación de la aplicación había avanzado lo suficiente, llegó el momento de pensar en la gestión de los datos que se manejaban en dicha aplicación. Por lo tanto esta tarea se basa en el estudio del sistema de gestión de bases de datos MySQL, que fue el elegido en un primer momento para tal fin.
- **Tarea 6:** posteriormente al estudio de MySQL, en esta tarea se llevan a cabo los conocimientos adquiridos y se implementa dicha base de datos con las tablas necesarias para gestionar correctamente los datos que se manejan en la aplicación.
- **Tarea 7:** como se ha explicado anteriormente en el apartado **2.2.4** se decidió implementar finalmente la base de datos basándose en SQLite en lugar de MySQL. Por lo tanto esta tarea corresponde con el aprendizaje necesario para utilizar SQLite bajo el lenguaje de programación Python. Como se observa en la tabla **A.1** esta tarea es más corta en el tiempo que el aprendizaje de MySQL, ya que los conocimientos previos ayudaron en la presente tarea, pues la implementación de la base de datos no difiere demasiado de usar uno a usar otro.
- **Tarea 8:** este periodo corresponde con la implementación de la base de datos utilizando el módulo SQLite3, que es el que corresponde para implementar una base de datos utilizando SQLite mediante lenguaje de programación Python.
- **Tarea 9:** esta tarea se correponde con el periodo de aprendizaje del lenguaje de LaTeX, necesario para la redacción posterior de la memoria.
- **Tarea 10:** periodo que corresponde con la redacción de la presente memoria del trabajo de fin de grado.

- **Tarea 11:** esta tarea se corresponde con la realización de la presentación mediante diapositivas que se empleará en la defensa del presente trabajo de fin de grado.
- **Tarea 12:** la última tarea se corresponde con las horas de tutorías. Esta tarea no ha sido incluida ni en la tabla A.1 ni en el diagrama de gantt que se mostrará posteriormente en la figura A.1.

La realización de las tareas normalmente ha sido secuencial, es decir han llevado un orden, aunque esto no quita que en muchas ocasiones se haya tenido que volver hacia atrás y modificar una tarea ya realizada, con los efectos sobre las otras tareas que esto conlleva. Es decir se podría decir que se ha seguido un modelo de ciclo de vida en cascada con retroalimentación.

También cabe destacar, aunque las tareas aparezcan reflejadas en la tabla con fecha de fin, que realmente hay algunas tareas que no han finalizado hasta llegar al fin de la elaboración de todo el trabajo de fin de grado. Tareas como el aprendizaje de Python o Pygame. Estas son tareas que en el principio tuvieron un periodo de dedicación exclusiva a dicho aprendizaje, pero realmente durante toda la etapa de desarrollo de la lógica del juego y de la interfaz gráfica se han seguido realizando estudios sobre Python y Pygame, ya que se trata de tareas continuadas y van surgiendo dudas y nuevas ideas según avanza la aplicación.

En la **Figura A.1**, se muestra el diagrama de gantt con la planificación de cada una de las tareas de las que consta el presente trabajo de fin de grado.

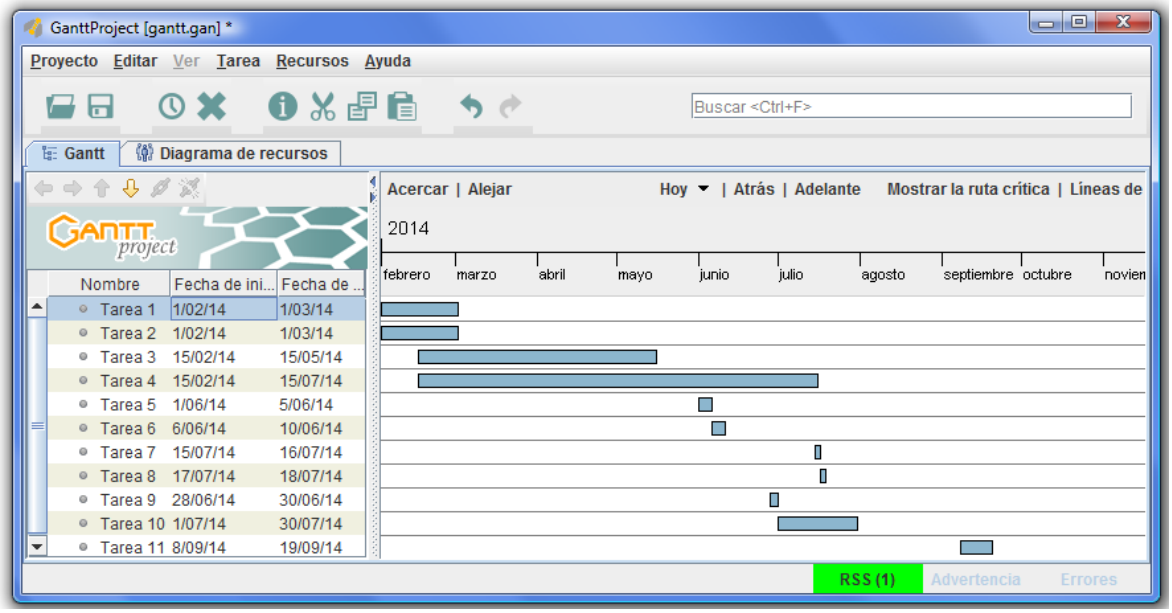

Figura A.1: Diagrama de gantt del trabajo de fin de grado.

A.2. Presupuesto

En este apartado se calcula el presupuesto aproximado que corresponde a la realización del presente trabajo de fin de grado. El cálculo del presupuesto se puede hacer en referencia a dos aspectos principales, que son las horas de dedicación empleadas en la realización de cada tarea y el coste del material empleado.

- **Horas empleadas en cada tarea:** en la tabla **A.2** se muestran las horas aproximadas que se han empleado en cada tarea de las que consta la realización del trabajo de fin de grado.

Tarea	Horas
1	66
2	66
3	198
4	238
5	24
6	24
7	2
8	1
9	6
10	120
11	30
12	15

Tabla A.2: Horas dedicadas a cada tarea.

Además podemos agrupar las tareas en cuatro grupos, para tener una mejor visualización de las horas dedicadas a cada gran parte del trabajo de fin de grado. Dicha agrupación se muestra en la tabla **A.3**:

Grupo	Actividad	Horas
1	Documentación	164
2	Desarrollo de la aplicación	461
3	Redacción de la memoria	120
4	Presentación	30

Tabla A.3: Horas empleadas en cada grupo.

Se puede detallar la actividad realizada en los diferentes grupos citados en la tabla **A.3**, de la siguiente manera:

- **Grupo 1:** agrupa las diferentes tareas de documentación y aprendizaje de las herramientas que nos permiten desarrollar la aplicación y la memoria, como Python, Pygame, SQLite y LaTeX.
- **Grupo 2:** en este grupo se reflejan todas las actividades referentes al diseño e implementación de la aplicación, tales como la lógica del juego, la interfaz gráfica y el diseño de la base de datos.
- **Grupo 3:** este grupo hace referencia únicamente a la redacción de la presente memoria del trabajo de fin de grado.
- **Grupo 4:** por último este grupo hace referencia a la elaboración de la presentación.

El total de horas dedicadas entre los cuatro grupos es de: 775 horas.

El periodo de realización del proyecto es de 7 meses, de Febrero a Septiembre y sin contar Agosto que no se trabajó. Estos 7 meses de trabajo equivalen a 28 semanas.

Por lo tanto, el número de horas semanales de trabajo es de:

775 horas / 28 semanas = 27.7 horas/semana

En la **Figura A.2**, se muestran las tablas correspondientes al coste mensual de contratación laboral para el año 2014 facilitadas por la Universidad Carlos III de Madrid:

COSTE MENSUAL DE CONTRATACIÓN LABORAL, SEGÚN TABLAS PARA 2014

	Titulado Medio			Titulado Doctor			
				MÍNIMO		MÁXIMO	
JORNADA	Salario bruto	Coste proyecto		Salario bruto	Coste proyecto	Salario bruto	Coste proyecto
17,5H	1045,91	1417,95		852,60	1155,88	1577,23	2138,27
22,5H	1344,74	1823,08		1096,20	1486,13	2027,86	2749,21
27,5H	1643,57	2228,21		1339,80	1816,38	2478,50	3360,14
32,5H	1942,40	2633,34		1583,40	2146,63	2929,14	3988,65
Completa (37,5H)	2241,23	3038,47		1827,00	2476,89	3379,77	4582,01
	Coste hora=	23,15		Coste hora=	18,87	Coste hora=	34,91

Figura A.2: Coste mensual de contratación laboral para el 2014.

Aproximando las horas semanales de trabajo a 27.5 y tomando como referencia el salario correspondiente a un titulado medio, observamos en la tabla que el salario bruto es de 1643.57 euros/mes.

Por lo tanto, el salario bruto recibido por los 7 meses de trabajo correpondientes a la elaboración del presente proyecto es de:

- 1643.57 euros/mes x 7 meses = **11504.99 euros**.

Por otro lado, también hay que tener en cuenta el coste de las horas de tutoría, que en este proyecto han sido 15 horas. En este caso se toma como referencia la tabla de salarios para un titulado doctor y la media entre salario bruto mínimo y máximo. Los salarios mínimo y máximo utilizados serán para un total de 15 horas y se obtendrán a partir de los datos presentes en la tabla y referidos a una jornada de 17.5 horas:

Salario mínimo = 730.8 euros ; salario máximo = 1352 euros

- Salario medio bruto = 1041.4 euros/mes, por lo tanto a la semana, que sería el coste de las 15 horas de tutoría sería: **260.35 euros**.

Por lo tanto, el coste total del proyecto referido a las horas empledas en el mismo es el siguiente:

- **TOTAL**: 11504.99 euros + 260.35 euros = **11765.34 euros.**

- **Material empleado:** en la tabla **A.4** se refleja el coste del material empleado en el desarrollo del presente trabajo de fin de grado:

Material	Coste
Ordenador	1300 euros

Tabla A.4: Coste del material empleado.

Ahora toca calcular la amortización del material empleado durante los 7 meses que ha durado el desarrollo del presente trabajo de fin de grado.

Para ello, aplicamos el coeficiente máximo para equipos informáticos, que es del 26 %. Este dato ha sido extraído de la tabla que ofrece cada año la agencia tributaria **[32]**.

En la **Figura A.3**, se muestra la tabla anteriormente mencionada.

Grupo	Elementos patrimoniales	Coeficiente lineal máximo (%)	Período Máximo (Años)
1	Edificios y otras construcciones	3	68
2	Instalaciones, mobiliario, enseres y resto del inmovilizado material	10	20
3	Maquinaria	12	18
4	Elementos de Transporte	16	14
5	Equipos para tratamiento de la información y sistemas y programas informáticos	26	10
6	Útiles y herramientas	30	8
7	Ganado vacuno, porcino, ovino y caprino	16	14
8	Ganado equino y frutales no cítricos	8	25
9	Frutales cítricos y viñedos	4	50
10	Olivar	2	100

Figura A.3: Tablas de amortización de la agencia tributaria.

Este coeficiente hace referencia a cuánto se amortiza el producto durante un año. De esta forma, aplicando el 26 % al coste original del ordenador:

- 0.26 x 1300 euros = 338 euros/año.

Teniendo en cuenta que el proyecto ha durado 7 meses:

- **AMORTIZACIÓN DEL MATERIAL** = 338 x 7 / 12 = **197.17 euros.**

Por último, se va a calcular el coste final de la implementación del presente trabajo de fin de grado. Para ello se suma el coste de amortización del material usado y el coste referido a las horas empleadas en el desarrollo del presente trabajo de fin de grado.

- **COSTE FINAL**: 11765.34 euros + 197.17 euros = **11962.51 euros.**

Bibliografía

[1] Ley orgánica de protección de datos. http://www.inteco.es//Formacion/Legislacion/Ley_Organiza_de_Proteccion_de_Datos/. Consultado el día 25/07/2014.

[2] Ley orgánica de protección de datos. https://www.agpd.es/portalwebAGPD/canaldocumentacion/publicaciones/common/Guias/GUIA_SEGURIDAD_2010.pdf. Consultado el día 25/07/2014.

[3] Félix Etxeberria Balerdi. Videojuegos y educación - universidad del país vasco. http://campus.usal.es/~teoriaeducacion/rev_numero_02/n2_art_etxeberria.htm. Consultado el día 3/07/2014.

[4] James Paul Gee. *What Video Games Have to Teach Us About Learning and Literacy?*. Palgrave Macmillan, 2007.

[5] Juegos educativos para nintendo ds. http://www.juegos-educativos.net/juegos_consola/juegos_nds.html. Consultado el día 15/07/2014.

[6] Buzz! el mega concurso. http://www.game.es/Product/Default.aspx?SKU=048437. Consultado el día 9/07/2014.

[7] Aprende con pipo. http://educadortic.wordpress.com/2012/03/19/aprendiendo-con-pipo/. Consultado el día 9/07/2014.

[8] Alberto Martín. 9ª edición premio jóven - pasión por los videojuegos. *Fundación General de la Universidad Complutense de Madrid*, 2006.

[9] Alex. Especial de videojuegos en la actualidad. http://www.players4players.com/articulos/articulo.php?t=1018&pag=1, 2008. Consultado el día 5/07/2014.

[10] Wikipedia. Torque 3d. http://es.wikipedia.org/wiki/Torque_(motor_gr%C3%A1fico). Consultado el día 7/07/2014.

[11] Wikipedia. Unity (software). http://es.wikipedia.org/wiki/Unity_(software). Consultado el día 7/07/2014.

[12] Página oficial. Unity. http://unity3d.com/es/unity. Consultado el día 7/07/2014.

[13] Roberto Albornoz Figueroa (RCAF). Conceptos básicos para el desarrollo de videojuegos en 2d - sprites. http://www.losersjuegos.com.ar/referencia/articulos/conceptos_basicos, Enero 2007.

[14] Página oficial. Pygame. http://www.pygame.org/wiki/about. Consultado el día 7/07/2014.

[15] Java (lenguaje de programación). http://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n)#Cr.C3.ADticas. Consultado el día 10/07/2014.

[16] Eclipse ide. http://www.genbetadev.com/herramientas/eclipse-ide, Enero 2014. Consultado el día 11/07/2014.

[17] C sharp. http://es.wikipedia.org/wiki/C_Sharp#Metas_del_dise.C3.B1o_del_lenguaje. Consultado el día 11/07/2014.

[18] Microsoft visual studio. http://www.visualstudio.com/es-es. Consultado el día 12/07/2014.

[19] Python. http://es.wikipedia.org/wiki/Python. Consultado el día 11/07/2014.

[20] El zen de python. http://mundogeek.net/archivos/2007/07/02/el-zen-de-python/. Consultado el día 16/07/2014.

[21] Bases de datos. http://www.maestrosdelweb.com/editorial/%C2%BFque-son-las-bases-de-datos/. Consultado el día 17/07/2014.

[22] Bases de datos relacionales. http://www.jorgesanchez.net/bd/bdrelacional.pdf. Consultado el día 17/07/2014.

[23] Mongodb. http://www.genbetadev.com/bases-de-datos/mongodb-que-es-como-funciona-y-cuando-podemos-usarlo-o-no. Consultado el día 13/07/2014.

[24] Sqlite. http://es.scribd.com/doc/52882068/SQLite. Consultado el día 13/07/2014.

[25] Mysql. http://dev.mysql.com/doc/refman/5.0/es/introduction.html. Consultado el día 13/07/2014.

[26] Ieee ranking de lenguajes de programación. http://dataconomy.com/ieee-ranks-programming-languages-java-comes-top/. Consultado el día 11/09/2014.

[27] Interfaz gráfica heidisql. http://fundamentosdebasededatos.wikispaces.com/file/view/Manual+de+Uso+Heidi+SQL.pdf. Consultado el día 24/07/2014.

[28] Texniccenter. http://sourceforge.net/projects/texniccenter/. Consultado el día 24/07/2014.

[29] Microsoft visio 2010. http://office.microsoft.com/es-es/visio/. Consultado el día 24/07/2014.

[30] Pynsource. http://code.google.com/p/pynsource/. Consultado el día 24/07/2014.

[31] Ganttproject. http://gantt-project.uptodown.com/. Consultado el día 24/07/2014.

[32] Tablas de amortización de la agencia tributaria. http://www.agenciatributaria.es/AEAT.internet/Inicio_es_ES/_Segmentos_/Empresas_y_profesionales/Empresarios_individuales_y_profesionales/Rendimientos_de_actividades_economicas_en_el_IRPF/Regimenes_para_determinar_el_rendimiento_de_las_actividades_economicas/Estimacion_Directa_Simplificada.shtml. Consultado el día 14/09/2014.